Spring 5-1-2011

GrowndSwell iPhone Application Development

Arti Chandok

Recommended Citation

Chandok, Arti, "GrowndSwell iPhone Application Development" (2011). *Syracuse University Honors Program Capstone Projects*. 252.
https://surface.syr.edu/honors_capstone/252

GrowndSwell®

GrowndSwell iPhone Application Development

A Capstone Project Submitted in Partial Fulfillment of the Requirements of the Renée Crown University Honors Program at Syracuse University

Arti Chandok

Candidate for B.S Degree
and Renée Crown University Honors

4/2011

Honors Capstone Project In: ________Computer Engineering

Capstone Project Advisor:________________________________
Shiu-Kai Chin

Honors Reader:______________________________________
Michael J Fransciello

Honors Director:_____________________________________
James Spencer, Interim Director

Date: ____________________4/27/2011____________________

Disclaimer The information contained in this document is subject to change without notice. GrowndSwell, LLC. shall not be liable for costs accrued due to technical or editorial omissions or mistakes in this document.

Abstract

GrowndSwell is a location-based social networking service developed to primarily solve the problem of trying to text several friends in order to convey a simple piece of information: “What are you planning to do today?” This capstone project is focused on developing an iPhone application as a tool for users to convey this information in an efficient and fun manner. The problem requires several tools to capture the market through development of phone applications that are compatible on several platforms and creating an attractive website for brand awareness.

This capstone project has focused its attention on the product development of the iPhone application. It is targeted to developers and product managers who would be enhancing these tools to easily adapt to current processes and construct upcoming user interface designs and features. The capstone includes a comprehensive business plan for general business exposure of the services and products the company offers. It includes flowchart and user functionality documentation that are focused on the design aspects and the user interfaces and operation of the application. And lastly, the original code developed for the iPhone application.

The problem solving approach in the development and design of the iPhone application follows a typical top-down approach; it was created taking the larger features, namely the going-out, staying-in, and studying agendas and was then concerned with the details involved in each feature. Our team made various transitions in the design and features however with the waterfall approach, no code was developed until the demo screens were confirmed. This made the software development process with the analysis and design easier. The testing, and implementation are done simultaneously hence not all features all implemented however the completed features are tested on the emulator.

The software development and design are completed using vendor applications. The design and flow implementation used creative design adobe software such as Illustrator and Photoshop. Developing and testing the iPhone application were possible using Apple iOS SDK that are available online free of cost: (http://developer.apple.com/).

While the capstone provides a variety of documents to preview, the focus is primarily on the developers and product managers who are able to obtain a general overview from the business concepts, to the details involved in the design and development of the iPhone application of GrowndSwell.

Table of Contents

Acknowledgments

I would like to thank the GrowndSwell team, including Michael Lundberg, Nick Mancini and Scott Levy, who had initially approached me with this idea when I acknowledged on collaborating to develop their application. There are several components in this paper excluding the source code that are a collaborative effort of the team. This includes the business plan, the screen designs and features, and the presentation provided.

For integrating this location-based application with third-party vendors and external social sites, source code for the web scripts has been provided by a hired consultant, Surendra Kumar.

And lastly, I would like to acknowledge my honors advisor, Shiu-Kai Chin, honors reader, Michael Fransciello, the honors deputy director, Erick Schmidt, and the Renee Crown Honors Program for their tremendous support and continued faith in the completion of this capstone.

Advice to Future Students

This paper focuses on the software development process including the analysis, design and implementation and testing of the iPhone application. University courses focus only on specific aspects of the software development process either towards software specification or towards phone implementation. This project is a great way to learn and experience the entire developmental process and overview the minimum requirements that should be included prior to practicing, perfecting and approaching to solve a problem.

There are several podcasts containing online classes, books for references, and forums online to help answer technical questions. I would definitely recommend this experience.

Introduction: GrowndSwell's Vision

At GrowndSwell, we believe that social media interaction should not be limited to a digital network, but be a means to facilitate real social interactions. GrowndSwell is an itinerary-based app that helps users efficiently create their social agendas that can be published to your friends network. Setting up your status brings users the flexibility to attend and share new events. Popular social agendas once posted are visible to friends. The platform encompasses all aspects of a college student's social life including going to bars, restaurants, and clubs; staying in to watch movies and shows or play games; and study with friends or classmates. The simplicity of the three features allows a user to view and attend trending venues, watch movies and shows with friends, and find study buddies in their vicinity. Our app will organize social activity in real time allowing users to make plans for a more social evening.

Potential competitors include location-based social networks that allow friends to explore local cities and entice users to enter venues through group discounts. Foursquare, Gowalla, HurricaneParty and Ditto are examples to name a few.

Giving users the flexibility to influence others, makes group invites, group discounts and ultimately bringing groups of friends to new venues a lot more effective.

This contrasts vastly from Foursquare, where updating your current social status has limited influence on your friend's social events.

GrowndSwell is effective as it publishes today’s status, and is unlikely to overwhelm users with uncertain events. Ditto, is a similar service, that falls short of maintaining a time status and inefficient with no availability of trends.

GrowndSwell gives qualitative information to users that can create and disrupt social trends, be real-time and still provide efficient means to follow and search trends.

Business Plan

By:
Michael Lundberg
Nick Mancini
Scott Levy
Arti Villa Chandok

Section I: The Industry

GrowndSwell is a location-based social networking service that focuses on the organization of people rather than the organization of places. Our service will inform users on a mobile scale of their friend's nighttime agenda in order to enhance social interaction. Our industry is Internet Publishing and Web Broadcasting and Web Search Portals, NAICS code 519130, which, "comprises establishments primarily engaged in 1) publishing and/or broadcasting content on the Internet exclusively or 2) operating Web sites that use a search engine to generate and maintain extensive databases of Internet addresses and content in an easily searchable format (and known as Web search portals)" (NAICS, 2007). This industry encompasses a large range of websites from search engines to social networking, which includes a wide demographic of people. There are many segments involved in this industry, but the segment we fall under is web communities. Although the industry is filled with many different companies that all provide different services, we share with our competitors a common goal, which is to better connect people and provide accurate and useful information.

Social networking accounts for 13.7% of this industry. There are approximately 50,000 social networking websites in the United States (IBISWorld, 2011). This industry contains both private and publicly held companies with over 50% controlled by five major technology companies, but there are also many companies with small market share and high profitability as well. Frequent in the industry is the acquiring of small companies by larger established technology firms. Despite these acquisitions the industry is still very competitive due to the minimal barriers of entry and the low start-up costs, which in turn means that new websites appear everyday that pose competition to those already out there. Ultimately this means that the industry contains more small businesses than big ones.

As of 2010 the growth rate of the industry was at 24.9%, down 1.6% from 2009, but up 7.7% from 2008. However, it is projected that the growth rate will decline over the next three years down to 6.2%. The industry revenue is projected to average an annual rate of 6.8% up until 2016.

The success of a website within the Internet Publishing and Web Broadcasting and Web Search Portals industry depends on multiple factors. Among these things is creating a website that allows people to connect with each other, allowing businesses to connect with people, and the reliability of the website. Visitors will likely become loyal to websites that can offer these things. Bigger websites often have the upper hand in this area as they have been around for longer and have the money to spend on servers, security, and other technologies to keep the websites functional and reliable.

Section II: The Company, Concept, Products and Services

The Company and the Concept

Of the 67% of people in the United States who use web communities, 19% (eleven million people) use them on their mobile devices (Nielsen, 2009). The main demographic of people that frequent these web communities are between 18 and 49 years old.

Most social networks today ask their users to answer one distinct question, “What are you doing?” While this information is relevant for any Web 2.0 based company, it doesn’t provide much information for its users. When a user provides information regarding what they are doing at that exact moment in time, it does not allow for any sort of planning ahead or social gathering. At GrowndSwell we believe that it is more important to get to know your friends away from the computer or any sort of GUI.

The Products and Services

Our mobile application will allow for users to publish social agendas or itineraries so that they may easily coordinate their plans in order to engage with their friends in a physical environment. GrowndSwell is not a medium for establishing who you are online, but rather what you are going to do. It is a way of planning social gatherings and interactions. With the rapid expansion and progression of social media and mobile applications, we feel that the digital world has not provided a proper interface to foster physical social activity. Rather than talking about previous events or reminiscing about high school days, our application enables its users to plan social events ahead of time and do so in an expedited fashion. Our service ultimately provides our users with real time statistical data of their friend’s agendas and the social scene surrounding them.

With GrowndSwell users can create, revise, and publish social itineraries that can be viewed by all of their friends. Users will be able to see trending locations (locations that experience the most visitors) in order to gain a better understanding of the social scene. Users can reference a statistical map of their area to plan an event or series of events based of the social scene surrounding them. While a check-in feature is quintessential to the success of our network, it is the ability to recognize and then weigh your options and to plan future events that establishes dynamic user-to-user interaction that will ultimately give us the upper hand amongst our competitors.

With GrowndSwell, businesses will have the opportunity to attract customers by sending out “Swells”. “Swells” are messages that a bar or restaurant can send out to GrowndSwell users alerting them that if a certain amount of users arrive at the location by a certain time then they will receive some sort of discount. However, we take the Groupon approach to discounts and allow the businesses to establish that there will be no discount unless a certain number of people show up. This feature allows the venue to secure that it will get a certain amount of business that night. Also, the more friends the user shows up to the venue with, the better the discount the user and the friends receive. The point of this is to encourage people to go to the venue in order to help increase business and revenue for that venue. When GrowndSwell first starts this feature it will be free for businesses, however as GrowndSwell expands and grows we will take a percentage of the money the business makes from the “swell” they sent out or we will charge them a fee for sending out the “swell”.

Entry and Growth

GrowndSwell will take the form of an LLC in Connecticut in the summer of 2011. The first major step we have taken is designing each page of the mobile application and making sure that they all flow together to allow smooth navigation. The second step we have taken so far was to outsource the programming to a private programmer in India. In April of 2011 we will apply to release it on the Apple App Store as an “Alpha” product to Syracuse University students. Anyone will be able to download the application from the App Store, however only those with an syr.edu email address will be able to use the application.

GrowndSwell will release the application in “Beta” form in the summer of 2011 to iPhone, iPad, and iPod Touch users as well as expand the product to the Android operating system. Also during the summer we will be talking to venues about having them use our service to attract customers to their establishments. In the fall of 2011 we plan on taking the application out of “Beta” and releasing it to the public. In March 2012 we will release version 3.0 to New York City. Over the next year we will expand our application to different cities such as Boston, Washington D.C., and Philadelphia. GrowndSwell plans to expand to the major cities on the West Coast and in Western Europe once we have established a large user base. We believe that with the unique features that we offer to both businesses and users, we will gain many users and substantial capital.

Section III: Market Research & Analysis

Market and Customer Overview

Our target market exhibits characteristics similar to those of a socially active college student. We feel that this type of user would provide meaningful content to our network. While GrowndSwell is a medium through which to plan events, it also incorporates the studious aspects of a college career (e.g. studying). When a user signs up for our service they have an opportunity to register their courses. This way when a user is not feeling outgoing or must study for an exam, they may inform their friends and classmates that they are studying and for which class. This benefits both GrowndSwell and it's users. While the users are constantly publishing information to the network, they are providing relevant information to their friends and classmates so that they may meet up or study together.

Market Size & Market Trends

It has been estimated by The Neilsen Company that one out of two Americans will have smartphones by Christmas of 2011. These figures along with the rapid expansion of mobile applications will provide a foundation for our success. Similar mobile networks, such as Foursquare, have reported that they are increasing by roughly one hundred thousand users every ten days. If you bring the proper network together and implement it on a mobile scale, the potential for a venture to thrive is insurmountable. There is no denying the demand or market for mobile applications.

When introducing a web 2.0 product to new users there has to be content that engages the user in order for the site to succeed. At GrowndSwell we feel that a major flaw in most networks today is that they lack the opportunity to network in a physical sense. When introducing our product we will implement a login through Facebook, our users will be able to seamlessly transfer their contacts and other information.

Whether through Twitter, Facebook, FourSquare, or even text messaging, college students are routinely publishing their information online to engage with one another. It is almost as if there is a constant connection between the users, the user's mobile device, and the social network.

Bob Hazlett from onehalfamazing.com has stated that of the most common social networks today, 45% of their users are between the ages of 18 and 34. This demographic represents both college student and young adult who exhibit the qualities that our network desires. GrowndSwell would provide a medium for this demographic to display their social activity for all of their friends to see. This information would allow for our users to set up study sessions, movie nights, parties, and other types of social gatherings.

As our network begins to develop traffic and user interactivity we can then monitor our users' activity and provide local businesses with this data. Local businesses can use this information to help them set up specials and deals ("swells") that allow for further social interaction to take place. By monitoring consumer trends, GrowndSwell can understand the dynamics of the social scene in a given area and redistribute that information to help local businesses thrive.

User Demographics

The dynamics of a college environment are vastly similar to the information published on our network. A college brings thousands of students together in one location and allows for these individuals to interact. However the means for interaction and understanding the social scene can be somewhat of challenge. With GrowndSwell, students will be able to publish or advertise their social agendas for all of their friends to see. They can set up or plan events at local venues and then advertise these social gatherings to their immediate friends.

Not only does our network help bring people together it also brings people to businesses in larger groups. In a collegiate environment, students are often many miles away from family so meals are often spent with close friends. The implementation of a social itinerary network in a college town would essentially create a medium that would allow for local businesses to reach out to the mass of students. While our network would serve the initial purpose of bringing people together in distinct locations, it would also allow for businesses to bring people in by the masses.

Market Segmentation and Targeting

The location-based social networks popular today operate on the premise of exploration, challenges, and integration of other networks. Companies such as Foursquare and SCVNGR both have similar location-based networks that implement a check-in feature that notifies the service of your whereabouts. Foursquare has a business minded approach; upon checking in users can receive discounts at local stores and venues. SCVNGR on the other hand, provides a check-in feature that issues challenges to encourage exploration of one's city.

Gowalla is another similar location-based social network. Just like Foursquare and SCVNGR, Gowalla's emphasis in on exploring your surroundings. However, Gowalla also allows users to share past travels and experiences with their friends.

Competitive Edge

The current landscape of location-based social networking appears to be segmented by exploration and challenges versus business and consumer benefits. At GrowndSwell while we feel that these are valuable services to a location-based app, they do not encourage user-to-user interaction. As a social network or Web 2.0 company it is imperative that users have an appropriate medium for interaction. With GrowndSwell's itinerary based social network our service allows for check-ins at local venues, user-to-user interaction, and user generated content.

Our application based social network operates on the publication of a user’s itinerary or agenda. This information is highly valuable to the user’s immediate friends. Upon the publication of an itinerary or agenda, other users can comment, copy, or suggest revisions to the itinerary in order to promote social interaction. Our service goes far beyond the comment on a picture or a status update. The publication of an itinerary promotes social gathering away from the desktop computer. While services similar to foursquare implement real-time location updates, there is yet to be a way to plan an event or check-in.

It is ultimately impossible to determine the location of a friend or individual given the sporadic nature of human behavior. We believe that granting users the ability to publish their social schedules strictly to their friends would create a new medium for event planning and ultimately improve social interaction. No longer will college students have to ask everyone in their contact list what their plan is for the evening.

Market Share

Neilsen has also calculated that social media dominates the amount of time we spend on-line. In 2010 it was estimated that 22.7% of time spent online was for social networking. Sites such as Facebook and Twitter contribute drastically to these numbers. While GrowndSwell is considered to be a social network, we feel that social networking should not take time away from being social. The authors of cruchgear.com have estimated that the average teen spends roughly 31-hours a week online, which equates to roughly 7-hours a week of social networking. Our service will allow for up-to-date itineraries in real time, thus providing easy to access and relevant information. Our service will make it easier than ever to plan a Friday night out with your friends or a study session with a classmate.

On Going Market Evaluation

Taking into consideration the progression of the web and how and where we interact on it, we can begin to see the trends that form around it. In the 90's the first web authors started creating web pages. One person would publish thousands of words into an html document that would be read by potentially thousands of people. Now in 2011, there are multiple networks and pages that anyone can join and publish 140 characters at a time. The easier it is for people to publish information to a network has a direct correlation to how many people will join the website.

All of those involved in GrowndSwell firmly believe that the content produced by a collegiate demographic would ultimately provide a dynamic network of information. Users can access their friend's profiles to see their social activity. Users can precisely determine the course of their evening based on trending social locations. Our network would harness the social activity on hand held interface. Wherever the user has access to the Internet via their smartphone or hand held device, they can submit itineraries, review locations, browse friend activity, and receive discounts from businesses. Our easy to navigate user-interface exceptionally captures every aspect of planning or gathering information for a social event.

Section IV: The Economics of the Business

Revenue Drivers and Profit Margins

GrowndSwell will have not have any revenue drivers for the first year. During the first year we will be dealing with costs associated to creating and constantly updating the quality and content of our application. However after the first year our three main revenue drivers will be advertising, sales of the information we collect to third party database companies, and the fee or percent we get from businesses to send out "Swells". At this point in time we plan on making the application free for users.

Fixed and Variable Costs

During the first years of operations, GrowndSwell will have higher fixed costs than variable costs. Most of these fixed costs will be one time start up costs, incorporation fees, Apple App Store fee to release our application, as well as continuing monthly payments for server space and a domain name. The startup costs will be described later in this section.

Operating Leverage and its Implications

GrowndSwell is considered to be a high operating leveraged company because the amount of fixed costs is much higher than the variable costs. This high operating leverage means it will take longer to breakeven. However, once GrowndSwell breaks even we will generate high profit margins through the fees the venues will pay to send out “Swells”, advertising, as well as selling the data we collect about our customers to third parties.

Startup Costs

GrowndSwell has many one-time startup costs. These costs are listed as the following:

Domain Name	$35
Server Space	$100
Apple App Store Fee	$99
Incorporation Fee	$1400

Overall Economic Model

Profit Durability

GrowndSwell’s profit stream will vary. Profit will depend on the amount of users who download and use our application and website. Our three major profit streams are from advertising, fees from venues, and selling the data to third parties. The only thing that makes our profit stream weak is the fact that it all depends on the amount of traffic we receive on our website and the amount of people that download and use our application.

Section V: The Marketing Plan

Overall Marketing Strategy

GrowndSwell’s strategy is to capitalize and successfully disseminate our phone application/web site to prospective customers through already existing and established social networking medias such as Facebook and Twitter. We plan on employing various representatives from different college campuses to help broadcast our product effectively across the nation. We anticipate an upsurge in brand awareness through word of mouth to reach our ideal audience goal.

Our competitors or companies involved in a similar market as us are Foursquare and Groupon. Foursquare focuses on where you currently are, and Growndswell provides users with where you are going to be. Groupon is a website that offers discounts and is solely utilized online, where Growndswell channels users into various venues, and discounts are generated by a number of users being present in a specified location at a specified time. We will actively display the benefits or distinguishable features that exist between our company and our competitors, and try to mitigate the opportunity cost of switching from other products to Growndswell.

Growndswell will follow the same expansion trajectory as social networking companies prior. We will develop strong relationships locally by having strong promotional activities, instating educational systems, and working rigorously to expand our reach over Syracuse University students. We believe that exclusivity raises an eyebrow and makes the appeal to certain products increasingly enticing. The idea is for Syracuse University to be our pilot school, and then quickly begin to proliferate to other campuses nationwide through student representatives on other college campuses. Consequently, as we spread to encompass more college campuses and users outside of the college demographic, our sales will increase exponentially due to the increased number of businesses that we will be working with.

Pricing

GrowndSwell plans on selling the application though Apple's App Store for free. We believe that people are more willing to download applications when they are free. Also since most of our competition offers their applications for free we need to offer the same registration fee.

For businesses, we plan on making money by charging businesses a rate to be able to send out special deals (discounts) or "swells" to customers. The fee will be applied to businesses either by using a flat rate for sending out a "swell" for the night, or by taking a percentage of the revenue that company receives through broadcasting a special deal. The various undertakings will be interested in conducting business with Growndswell because incentivizing events on our service will generate an upsurge in traffic, customer loyalty, and company awareness. Other companies have similar successful pricing models, namely Groupon that is solely accessed online, where we differ by offering these deals on a mobile scale.

Selling Cycle:

The process of persuading a prospective user to alienate an existing service, and adopt a new one is very difficult. However, Growndswell offers unique features that are currently not found in any other established social networking services, which has the ability to directly confront and remedy an ongoing problem. We will aggressively inform our target audience through promotions and advertising how Growndswell can enhance environmental activities while squashing existing problems simultaneously. Once users in our market realize the advantageous implications of utilizing our service, it will be difficult to neglect.

Advertising and Sales Promotion

GrowndSwell plans on taking advantage of various existing social media websites to be able to get our brand name out there. We plan on creating a Twitter account and using it to advertise the release of our application and create hype. Also using Facebook we will create a GrowndSwell page where we can post updates about our application, offer information about our company, and request friends. Another way we plan on advertising GrowndSwell is by sending out mass emails through other university list-serves to help generate student recognition and knowledge on how to use our studying feature to enhance scholastic opportunities. A huge advertising vehicle we will be adopting is the employment of student representatives on other college campuses that will set up booths and educate users on the benefits and functionality of Growndswell.

Customer Service, Warranty, and Guarantee Policies

GrowndSwell will have a help page on the website offering aid to customers with any problems they may experience with the application. Also, there is a feature on our service for consumers to rectify or alert us if there are any imperfections or complications with the content that appears on our application. After the first six months we will have a 1-800 customer service number available during limited hours of the days, which has yet to be established.

We will also have a guarantee policy to make sure that all passwords and confidential information will remain secure. We also take pride in our strong privacy settings implemented in our service, enabling users to control or filter the people that are allowed access to view their personal content and the information they are broadcasting. If confidential information is leaked out we will have a lawyer on call to help us with any legal troubles.

Distribution

As mentioned earlier, initially GrowndSwell will apply for deployment through Apple’s App Store only. After a year, once we expand to encompass Android and BlackBerry platforms, we will be distributing the application through the BlackBerry App World and the Android Marketplace. We realized that the majority of App users are iPhone users followed by Android users so we designed the path of our product release to imitate this market trend.

SECTION VI: DESIGN AND DEVELOPMENT PLAN

Development Status and Tasks

GrowndSwell is currently developing an iPhone application prototype for alpha-version release on the Apple app store by beginning of May. The founders produced the screenshots and finalized features to be included in the prototype, along with the technical documentation. To expedite programming and involve experienced developers, we have negotiated a contract with a team of freelancer developers and a project manager in India with a development period of 2 months at a price of $3000. This includes development and testing of the front-end mobile application and back-end web server scripting and database handling. The prototype screenshots are provided in The iPhone alpha-version will be available on the iTunes app store for free download to selected users that are primarily college students during the alpha-testing phase. GrowndSwell has also applied for an enterprise account for $199 with Apple and would require 1 week from 21st March for release approval. GrowndSwell has two technical founders from Computer Engineering and Information Systems field giving us the necessary expertise to manage a team of developers and designers for rapid expansion in the development and design phase. It is critical however, that we acquire the necessary funding for hiring software engineers and user-interface designers to launch this idea on all smartphone platforms in a minimal time period.

The iPhone application will be used as an alpha-prototype available to Syracuse University students and businesses for feedback on existing features and to find other innovative revenue structures for beta-release. GrowndSwell aims to release other products that include:

- Phone applications for other platforms: Android, Palm, Blackberry, OVI for Nokia, and other devices.
- A featured website providing tools for integrating businesses with the end-users and other analytics for businesses to gain insight into consumer interests and behavior. This will enable revenue generation from Day 1, initially on a pay per use model and subsequently on subscription.

To continue expansion of our product and projected release in the 3 month summer period of 2011, Growndswell will hire a team of 4 full-time software developers and a project manager with an office in India. The company will also hire 2 engineers and 1 graphic designer in the US for maintenance, performance, innovation and quick updates required in the day-to-day operations of the current versions.

Difficulties and Risk

Opening an office in India as the development hub for our software applications has potential risk with communication, time difference and cultural barriers. Hiring an experienced and top quality project manager will be key to meeting deadlines. We propose to integrate improved communication technology into our US and India offices to eliminate most communication barriers. We will also hire employees on working on US time hours to eliminate time barriers. Our Founders have the necessary experience working with programmers and designers and a network with similar companies who have almost a decade of experience. This enables us to rapidly scale up and modify talent at the remote office. We will face risks with time to market throughout development stages as other potential start-ups come up with similar location-based-social media applications.

Product Improvement and New Products

Ongoing design and development of applications is key to our success. Our applications will always continue to be updated with newer versions. The iPhone prototype launch in Syracuse University will help the service be defined by our initial users, the college students. We will be leveraging their pride in the University to motivate them to give us feedback on our features as well as make suggestions for new ones to be added. We plan to devise a point system to reward users whose suggestions are incorporated into the future versions. Subsequently, we will be developing applications for other mobile platforms. Testing one product will help us minimize revisions to the design and aid in development of all subsequent mobile applications.

Costs

Overseas Costs

- The entire cost of hiring a team of 4 developers, a project manager, rental of office space as well as running expenses would be

$10,000/month.

- This is estimating an average salary of a software engineer with few years of experience of $1100/month.
- A project manager would cost us approximately $2000/month
- Overhead expenses including rental is estimated at $1000/month.

In-house Costs

- GrowndSwell will hire 1 graphic designer, and 2 developers with an estimated cost of $3000/month minimum wage with a 2-3% stock option.

Proprietary Issues

We would be developing our own software and would hold full proprietary rights to it.

SECTION VII: OPERATIONS PLAN

Operating Model and Cycle

Development, testing, design and deployment of our software applications and other web services are a critical part of GrowndSwell. Initially, new versions will be deployed every month and old model revisions will be made between 1-2 weeks. On the contrary, current versions in the market will not be updated but will require day-to-day fixes and updates. The two-split development processes will have different software teams located in India and the US, respectively. GrowndSwell will operate a larger technical team from India involved with development of new versions and a regional team in New York City that will deal with upfront issues involved with local users and businesses.

The marketing team will be located in USA. We plan to have a phased roll out starting with Syracuse University campus followed by a city wise launch with New York, Philadelphia and Boston in the initial phases.

Operations Strategy

Having a major part of our engineering team deployed in India, we will have competitive advantage with reduced salary, facility costs, and average higher working hours per employee located abroad. The New York team will be in close proximity to local business. It will provide better marketing and expansion for customers. Philadelphia and New York are both cities with many leading universities that will give the added benefit of reaching out to students through student ambassadors across campuses as a low cost way of adding to our user base.

Geographic Location

GrowndSwell will be running from the Tech Garden incubator, at no charge and minimal living expenses covered through the program, and will be located in downtown Syracuse at 235 Harrison Street, Syracuse, NY, during summer 2011. Following that, we will either continue to use the services and office space provided by this facility or move to New York City incubators. Locations such as 160 Varick Street charge $300 a month providing start-up incubator services. Advantages of New York city include proximity to business and end-users, good public transportation, and availability of hiring employees to work in the city. One of our founders is from New York City, so if we continue to be located in Syracuse, we can still carry out marketing activities at minimal cost. Our India office for a team of 5-6 developers can co-share offices with other companies known to the founders.

Facilities, Equipment and Improvements

Running offices in two locations will require high-end technology equipment across both offices. In India, expansion of engineers can be easily handled by moving to a larger office space with 10-15 employees. The US team can also quickly move offices to the New Jersey area where there is a prominent technology start-up leasing space available for teams of 10-12 people for rental of $1500/month with no deposit and long-term contract agreements. This will minimize risk moving to a nearby city while maintaining lower operating facility costs.

Legal Issues Affecting Operations

Growndswell will be using the expertise of our mentors from the University to initially do the legal paperwork for forming the company. We will be having terms of usage similar to other companies in the social media sphere for our users. We will also be signing contracts with local businesses for enabling them to use our platform to enhance their revenue. All these legal contracts will need insurance to protect us as the founders and the company for carrying out its business. We do not have the expertise for the same, but will be relying on our mentors and the incubator facilities for this.

Section VIII: Management Team

Management Team

Our management system at GrowndSwell is structured in a hierarchical manner. The premiere goal of this system is to ensure the stability and scalability of our product. We have separated our company into four different sections that allow for swift development and thorough management.

Board of Advisors

As a young start-up we have began consulting with many different leaders of the industry. Jeff Eddings, Phil Kaplan, and Jim Lisiak have been our primary consultants. _Jeff Eddings, a Silicon Valley mogul, has worked for a vast array of tech companies. He currently works at StumbleUpon as a Senior Project Manager. Prior to StumbleUpon Jeff excelled as a Senior Business Developer for Google. We feel that Jeff would be a very reliable resource on our board. His comprehension of the industry and varied connections would allow for us to scale our product in a properly expedited fashion.

Our other main contact for advising is Jim Lisiak, an ex-Microsoft GM. Jim has been in the field of computer science for nearly 40 years. After his many years with Microsoft Jim has ventured into start-ups and is now the Chief Technology Officer of Milyoni inc., a social media e-commerce company.

We have also been in immediate contact with Phil Kaplan, founder of Adbrite, Blippy, and fuckedcompany.com. Mr. Kaplan is a highly experienced entrepreneur that has smashing success with many of his ventures. All of us at GrowndSwell strongly feel that these individuals possess the skills and resources to drastically improve the success of our application.

Human Resource Management Needs

Our Human Resources department would be responsible for providing and maintaining the highest standards of working conditions. We feel that our working environment should make our employees feel like they never left home. In order to avoid any sort of confrontation, GrowndSwell will set standards for collaborative and safe working conditions.

Structuring the Offering or Proposed Deal

While GrowndSwell is a mobile application based start-up, we feel that rapid expansion is to our benefit. The sooner our company can provide service to our specified demographic, the sooner we can capture the market for mobile event planning. The only reasonable way for a company with little means to expand in such a way would be through venture capture and outside investments.

We feel that structuring an agreement in the VC companies in exchange for company shares and a return investment will grant our company a proper start-up. However, we at GrowndSwell feel that as a company we must stay true to our initial goals and intentions. We do not want to feel at all obligated or pressured to change the means of our business based venture capital input.

Section VIII: Financial Statements

Enclosed in Appendix A.

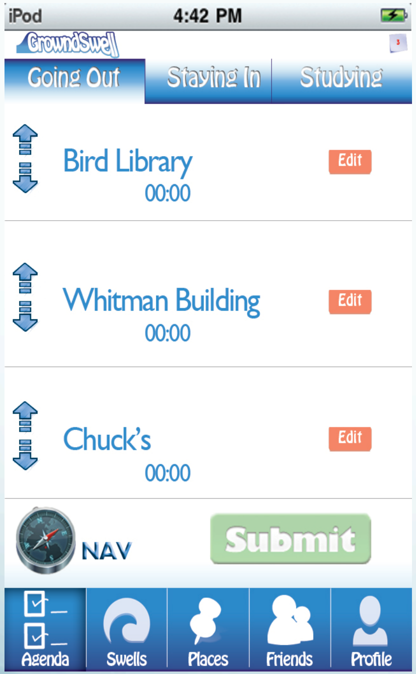
GrowndSwell

“surf the social scene with friends…”

What is GrowndSwell?

GrowndSwell is a location based social networking service that focuses on the organization of people rather than organization of places. Our service will inform users on a mobile scale of their friend's nighttime agenda in order to enhance social interaction. We also take a Groupon like approach to conducting business. Through our website businesses can set up incentives to entice users to come in and spend their evening at their locations.

GrowndSwell focuses on three integral aspects of a user’s night:

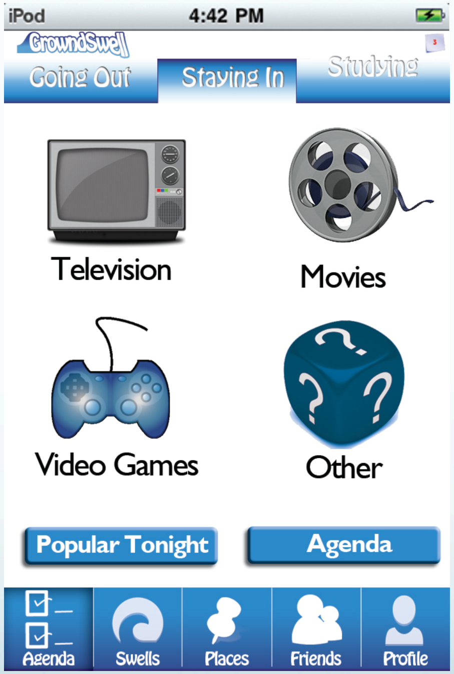


Bird Library 00:00

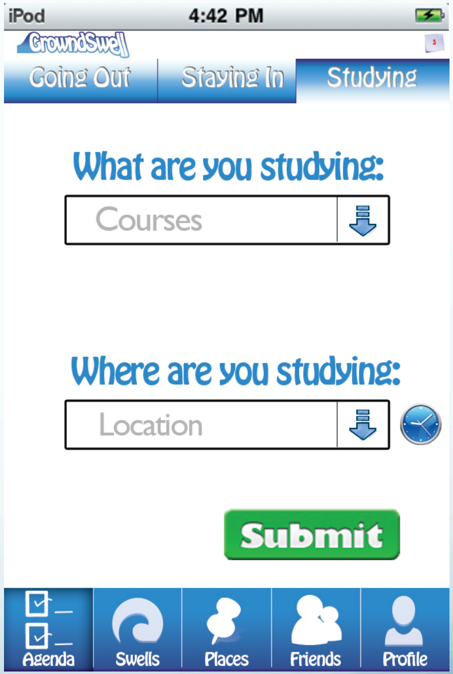
Whitman Building 00:00

Chuck’s 00:00

Going out agenda page



Staying in agenda page



What are you studying: Courses

Where are you studying: Location

Studying agenda page

Venue Page

To the right is a picture of a company's venue page. This page will display:

-Company's address

-Users currently at the venue

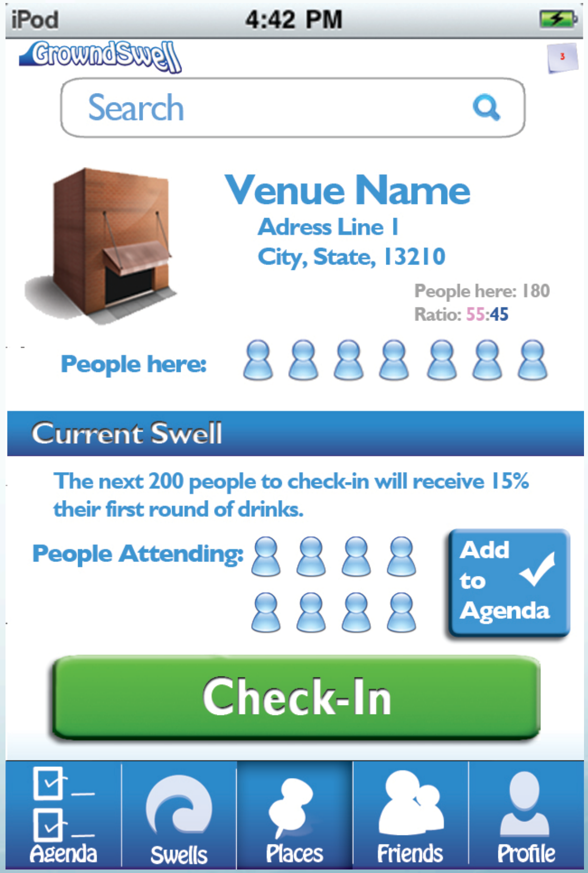
-Users who are going to be at the venue later

-A girl to guy ratio at the venue

-A "**swell**" if there is one for that night (a swell is a discount or special deal that a venue has for that day or night)

-An add to agenda button (A user clicks the button and it immediately gets added to their itinerary)

-Lastly, a check-in button (users can only click this if they are in close proximity to the respected venue)



The next 200 people to check-in will receive 15% their first round of drinks.

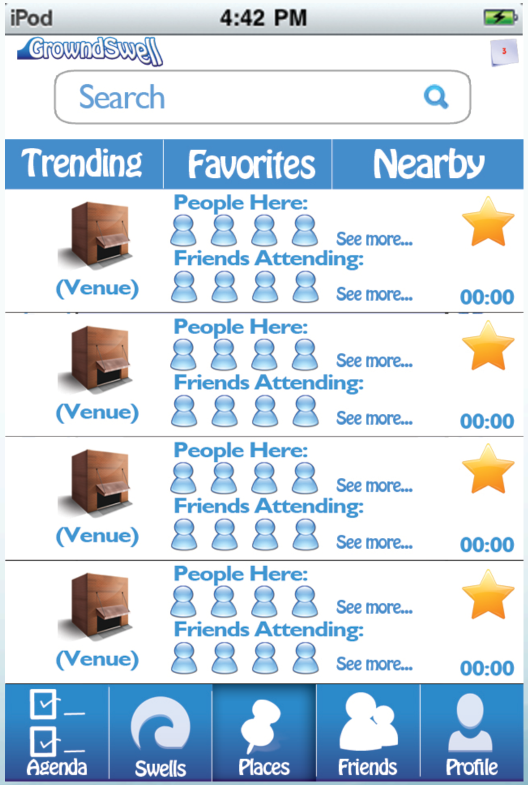
Places Page

The places page will show the users:

1) Trending- allows users to see what the most popular places will be for the night

2) Favorites- Click on the star next to any of your favorite venues, and it will immediately be listed on your favorites page.

3) Nearby- This will list all of the venues that are near you in a specific radius, in miles that you can customize on your setting's page.

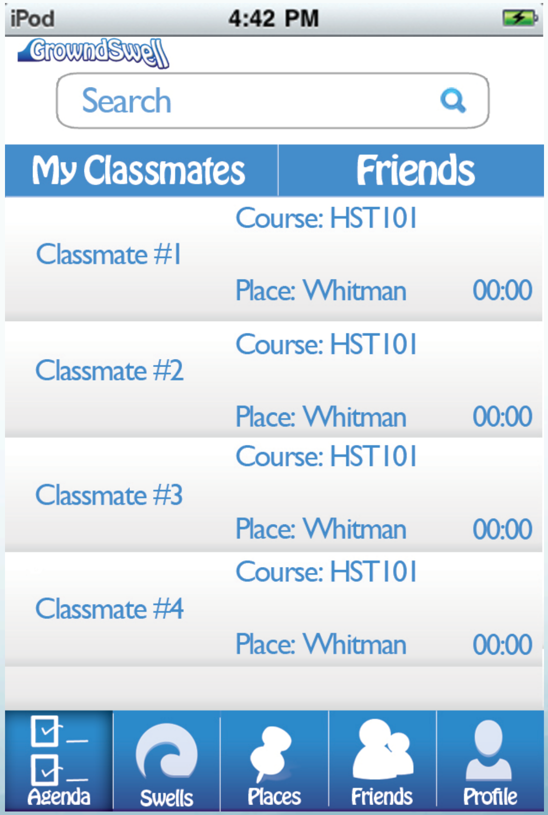


Finding Classmates

The unique aspect of using the studying feature is that it connects the user with classmates in his or her class to enhance academic opportunities.

When users register to be a GrowndSwell member, they will type in the courses they are enrolled in for the semester.

When they fill out their studying agenda for the night they will list what class and where they are studying, which will then bring up a concentrated list of classmates in order of similarity to that users agenda.



Classmate #1
Course: HST101
Place: Whitman 00:00

Classmate #2
Course: HST101
Place: Whitman 00:00

Classmate #3
Course: HST101
Place: Whitman 00:00

Classmate #4
Course: HST101
Place: Whitman 00:00

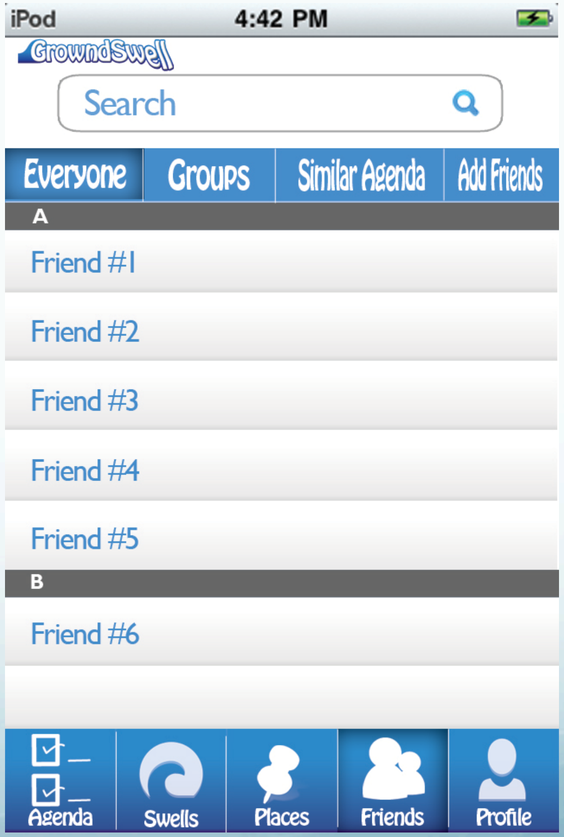
Friends Page

The friends page displays a user's friends in an easy and categorized way by showing:

-Everyone- all of the users friends

-Groups- customizable groupings of friends, making it easy to "follow" people.

-Similar agenda- this feature enables user's to see who in their network are partaking in similar activities for the night in real time.



Staying-in Movie Page

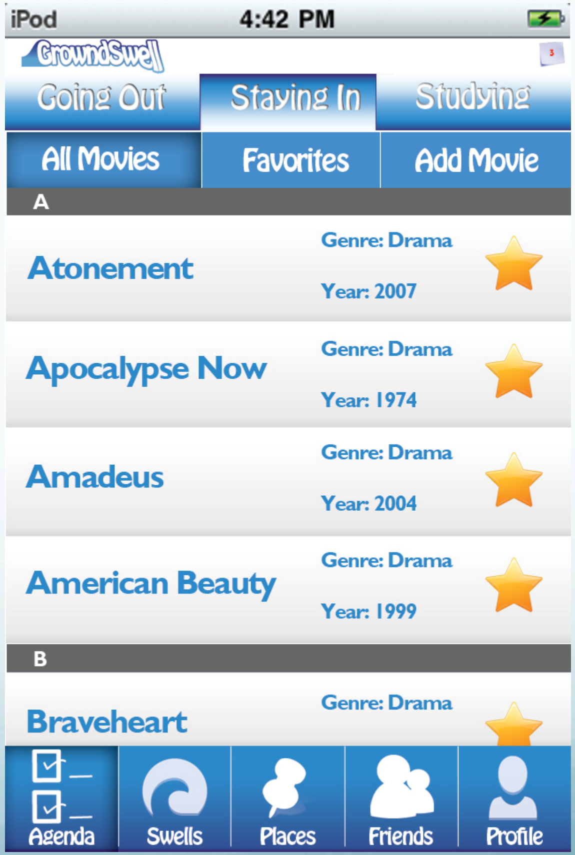
If a user clicks the movie button on the staying-in page it will bring them to this slide which includes:

-All movies tab- we will compile a lengthy list of movies from an open source database that will make finding a movie one wants to watch for the night easy.

-Favorites tab- a user can click on the star next to the movie title that will add that film to the favorites page.

-Add movie tab- If for some reason a user cannot find the movie he or she is trying to watch on the "all movies" tab, then they can add their desired movie to the list.

*Both TV and video game buttons on the staying-in page function similarly



Atonement Genre: Drama Year: 2007 ★

Apocalypse Now Genre: Drama Year: 1974 ★

Amadeus Genre: Drama Year: 2004 ★

American Beauty Genre: Drama Year: 1999 ★

Braveheart Genre: Drama ★

Profile Page

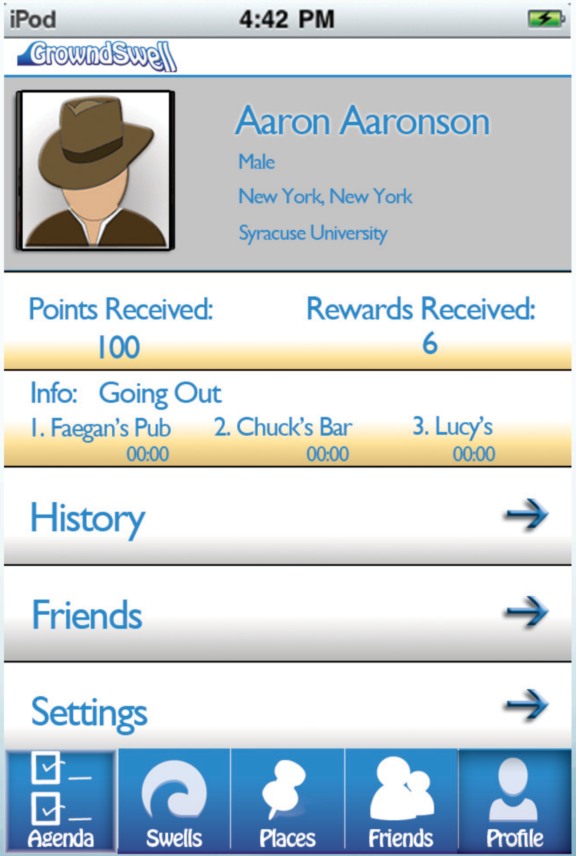
The profile page includes:

-Basic personal information

-Points/Rewards received- GrowndSwell will have a point system that will unlock premium discounts or rewards

-The itinerary the user created

-History- contains all of the venues that the user has checked into (frequency included), movies and tv-shows watched, video games played, and other activities over a set period of time.



iPod 4:42 PM

GrowndSwell

Aaron Aaronson

Male

New York, New York

Syracuse University

Points Received: 100

Rewards Received: 6

Info: Going Out

1. Faegan's Pub	2. Chuck's Bar	3. Lucy's
00:00	00:00	00:00

History →

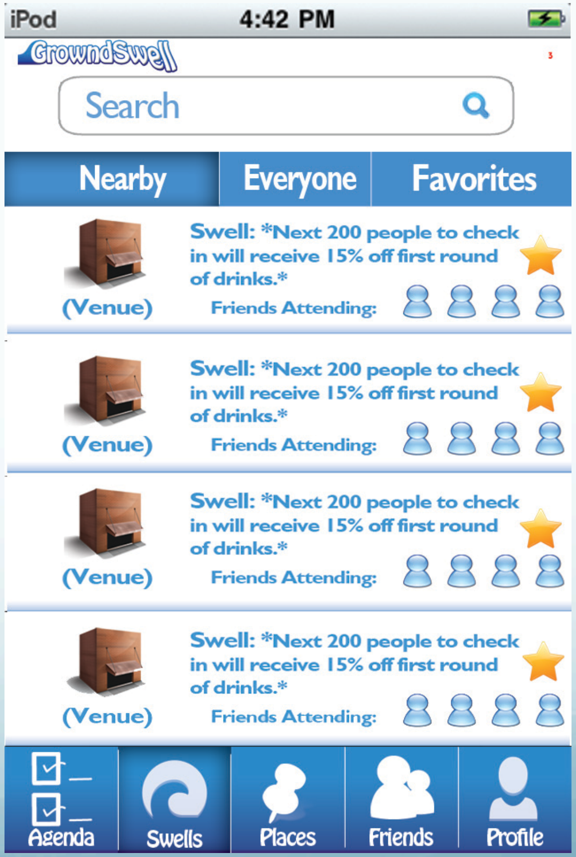
Friends →

Settings →

Swell Page

A swell is a discount or special deal that a company is having for a particular night.

This page displays all of the swells that are occurring nearby to the user, all swells that a user has access to, and a favorites tab that encompasses the users preferred swells that occurs at a number of venues.



(Venue) Swell: *Next 200 people to check in will receive 15% off first round of drinks.* Friends Attending:

(Venue) Swell: *Next 200 people to check in will receive 15% off first round of drinks.* Friends Attending:

(Venue) Swell: *Next 200 people to check in will receive 15% off first round of drinks.* Friends Attending:

(Venue) Swell: *Next 200 people to check in will receive 15% off first round of drinks.* Friends Attending:

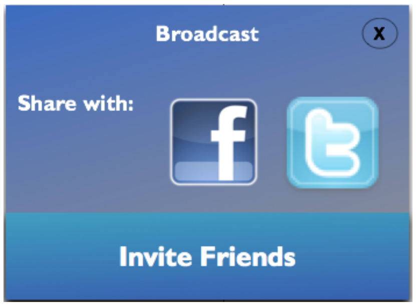
Functional Documents

By:

Arti Villa Chandok

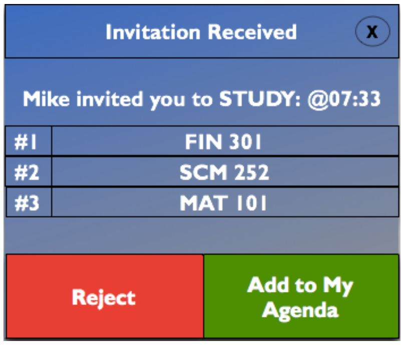
Broadcast

1. Facebook to post your agenda on wall Twitter to tweet your agenda
2. (the fb and twitter logos get highlighted)



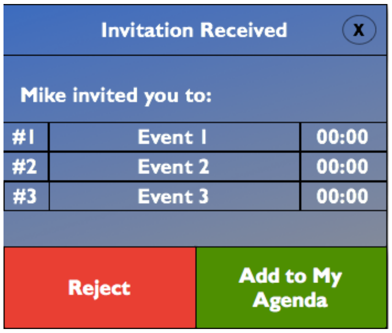
Invitation Received

1. User can highlight the rows to select which agendas to pick and thus replaces his entire agenda.
2. Add to My Agenda is faded until an event is selected.
3. Reject only removes the notice.
4. (Keep track of these Invitations in a table form inside the Inbox and the user can retrieve agenda these pre-set agendas from there for the current day)
5. Cross is used to remove the overlay from screen.
6. Reject sends a rejection notice to inbox.
7. The studying invitation is different but the going out and staying in have same '(Name) invited you to:' text



Mike invited you to STUDY: @07:33

#1	FIN 301
#2	SCM 252
#3	MAT 101

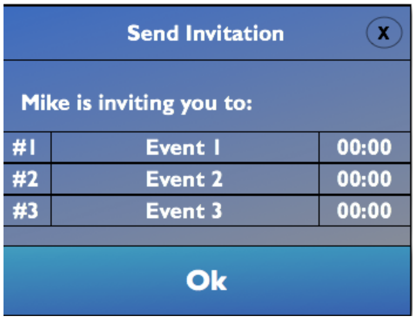


Mike invited you to:

#1	Event 1	00:00
#2	Event 2	00:00
#3	Event 3	00:00

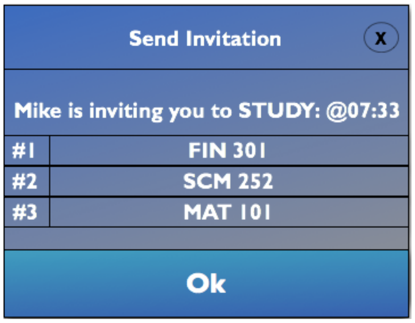
Send Invitation

1. Overlay the user gets when he 'Invite Friends' on the Broadcast Overlay.
2. Similar to Invitation Received, the user has to select which invitations he wants to send. The Ok is faded until at least one event is selected.



Mike is inviting you to:

#1	Event 1	00:00
#2	Event 2	00:00
#3	Event 3	00:00

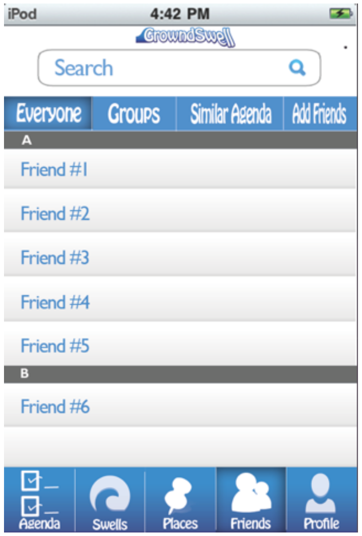


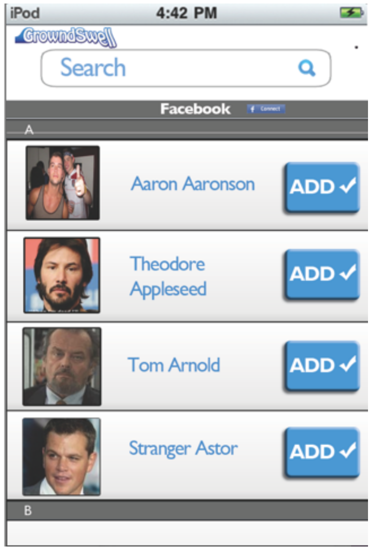
Mike is inviting you to STUDY: @07:33

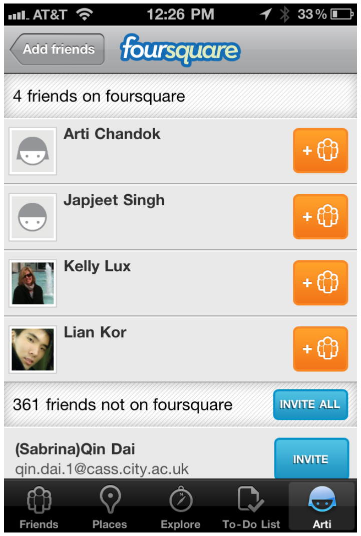
#1	FIN 301
#2	SCM 252
#3	MAT 101

GrowndSwell v1.0 Friend Screenshots

1. **General**
   a. Long list of friends has the character scroll bar viewable upon touch.
2. **Friends_Everyone**
   a. Displays entire list of your friends.
   b. Buttons: Message/Done, Edit
      i. Message: User selects this so they can highlight friends and press Done.
      ii. Edit: Minus sign to remove friends from list.
3. **Friends_Add_Friends**
   a. Add friends through fb, twitter, and address book.
   b. Friends that are on GrowndSwell (information can be found through fb, twitter connect) can be added. Friends that are not on the service can be invited via email. (Foursquare screenshot provided)

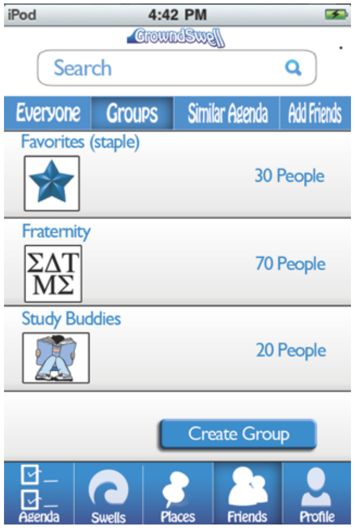


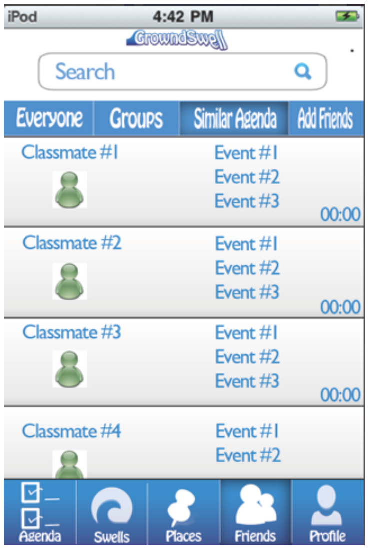


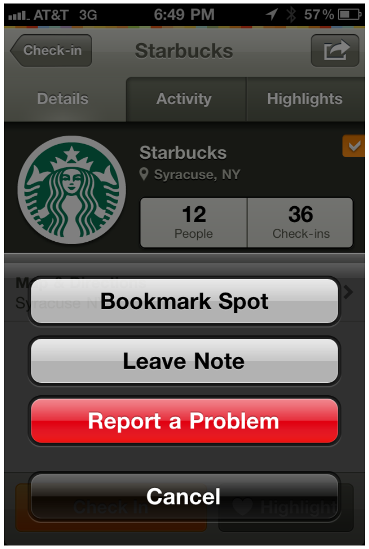


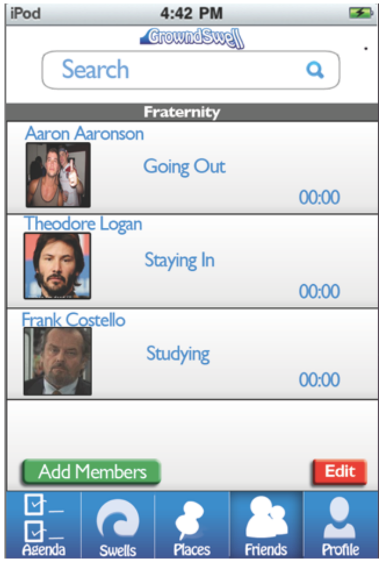
1. **Friends_Groups**
    a. Buttons: Create Group, Message Group
        i. Create Group: Can enter a group name and screen changes to Friends_Fraternity displaying name of the group as Title.
        ii. Message Group: User can select 1 group and screen changes to Edit_Message.
2. **Friends_Similar_Agenda**
    a. Friends are listed on a priority list based on similar courses and location, then similar location, and then similar courses and then the rest of the friends that have different courses and location if they have a current studying agenda.
3. **Friends_Fraternity**
    a. Buttons: Add Members, Edit, Message
    b. Add Members: Screen changes to Friends_Everyone to highlight friends and then press Done button to return with new friends added.
    c. Edit/Done button: Interchange on the spot. Edit button is to show minus sign to remove users and Done button is to return from edit mode.

d. Message button: Slides screen to ask user to ‘Message Friends’ or ‘Message Group’. Message Group changes screen to Edit_Message with all members being sent the message. Message friend shows a list of friends in the group for user to highlight and press Done button enter Edit_Message.

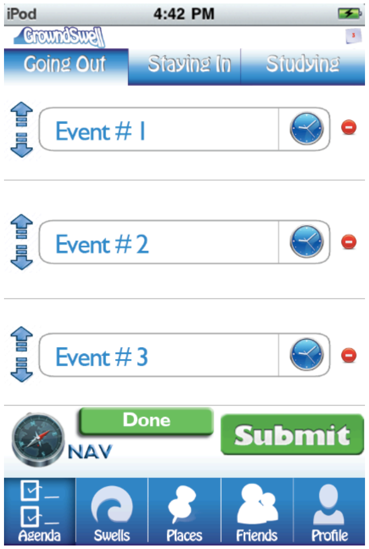


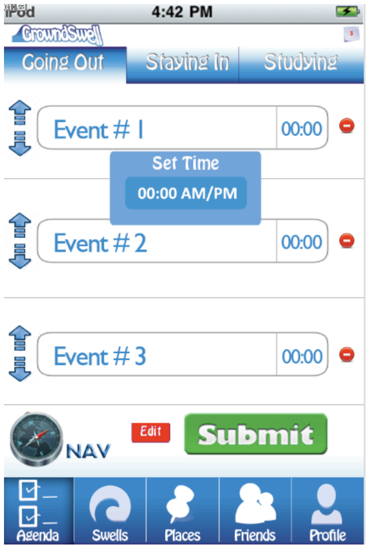


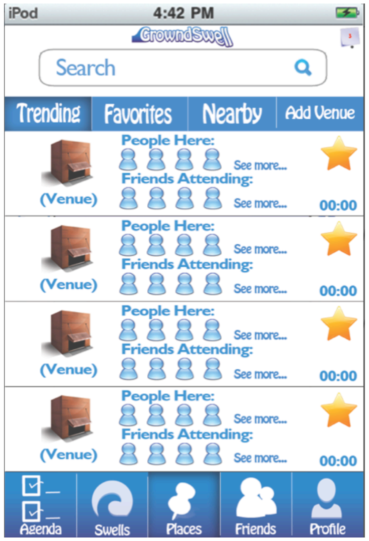




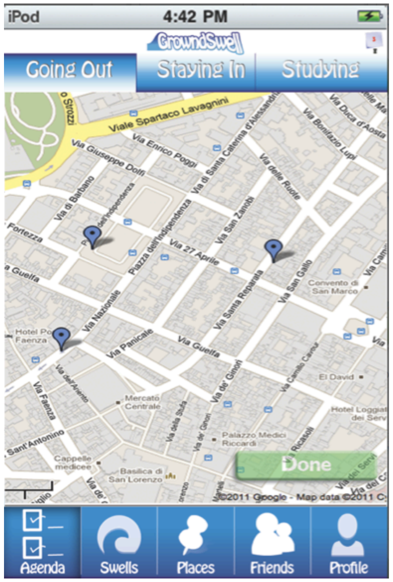
1. **Agenda_Going_Out_Event_Edit**
   a. Initial page when no events are selected by user
   b. User can select up to 3 events maximum
   c. Blue arrow: Re-arrange order of events by shifting with hand.
   d. Minus sign: Remove events and replace with ‘Add Event’ button.
   e. Clock button: Click to open default apple time setter and replace with a 12-hr time. Agenda_Going_Out_Set_Time shows how time is selected and the clock is replaced by the time. Screen looks like Agenda_Going_Out_Event_Time.
   f. Done button: Screen changes to Agenda_Going_Out_Summary.
   g. Submit button: Broadcast overlay is presented for user to post to fb, twitter, and invite friends. This button is translucent until done button is pressed.
   h. NAV button: Screen change to Agenda_Going_Out_Maps.
   i. Event# text field: Screen change to Places_Trending for user to select location from any of the places in navigation tab. Back button is available on all Places pages (Places_Trending, Places_Favorties, Places_Nearby, Places_Add_Venue) to come back to this page.



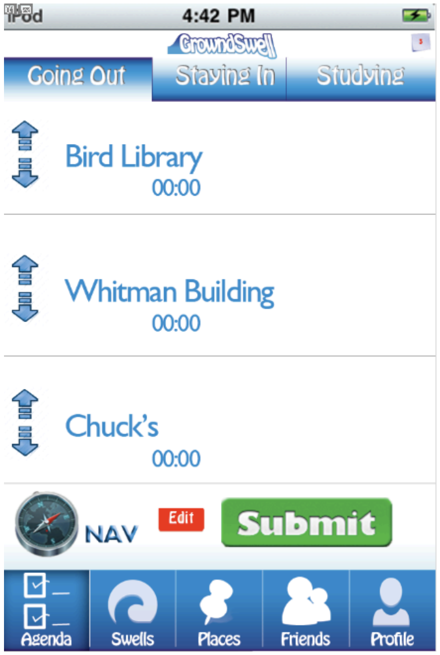


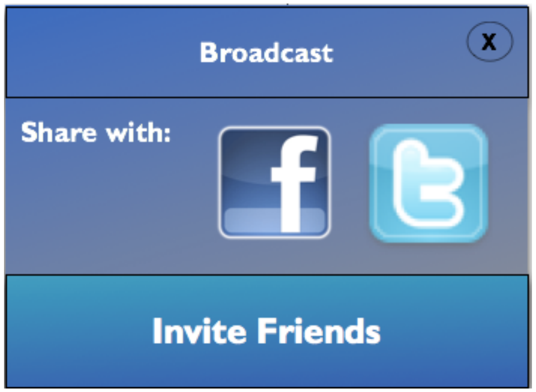


2. **Agenda_Going_Out_Maps**
    a. View annotated points on Google maps of locations selected on the Agenda_GoingOut_Events.
    b. Back button: Goes back to Agenda_GoingOut_Events.

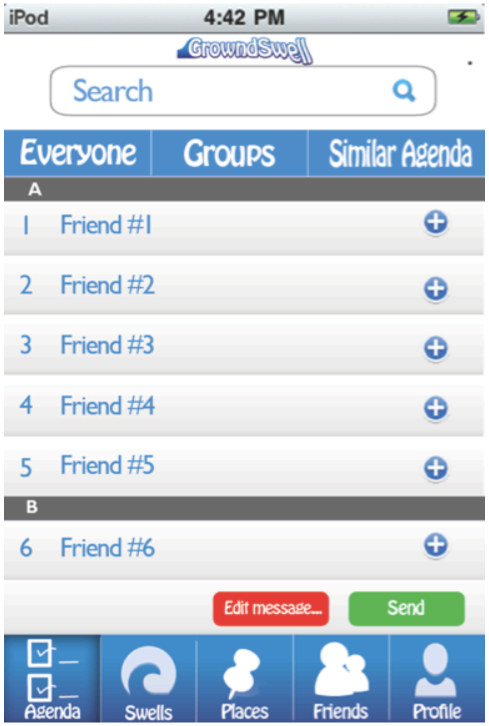


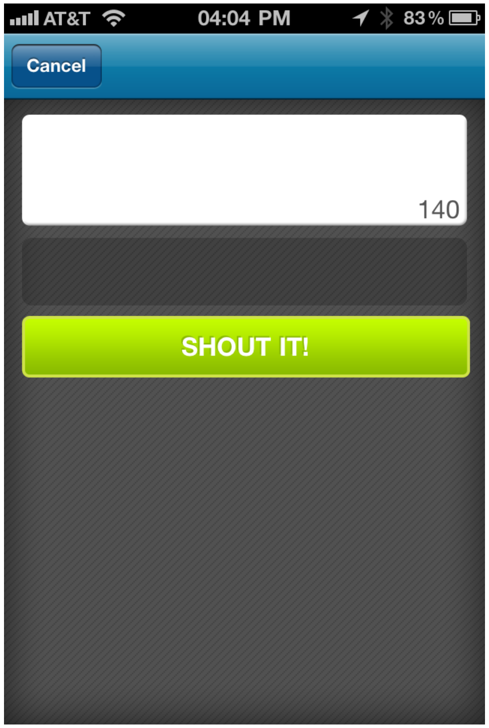
3. **Agenda_Going_Out_Summary**
    a. Cannot make changes to time and location of event on this view.
    b. Submit button: Pop-up with Broadcast overlay is displayed.
4. **Broadcast**
    a. User can post their agenda to fb, or as twitter update.
    b. Cross sign (x) to return to Agenda_Going_Out Summary.
    c. Invite friends button: Screen change to Friends_Going_Out_Everyone





5. **Friends_Going_Out_Everyone**
   a. Send button: Sends an invitation pop_up to friend to add the users agenda to their agenda with no message.
   b. Edit message… button: Screen changes to Agenda_Edit_Message. User can insert a personal message also.
   c. Select friends by highlight and send invitation using the two functions stated above.

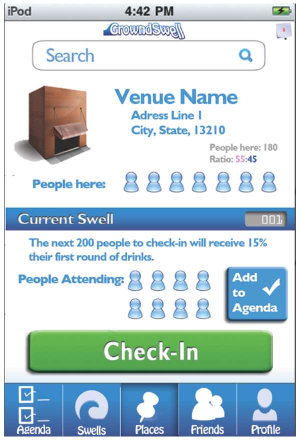




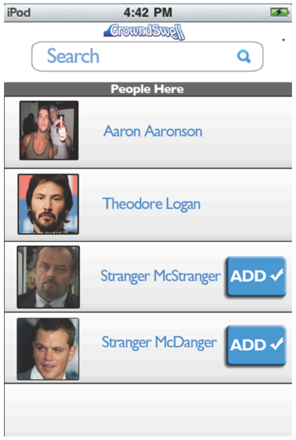
1. **General**
    a. Search bar (part of table list) is included in screen that contain a list of venues such as Places_Trending, Places_Favorites, Places_Nearby

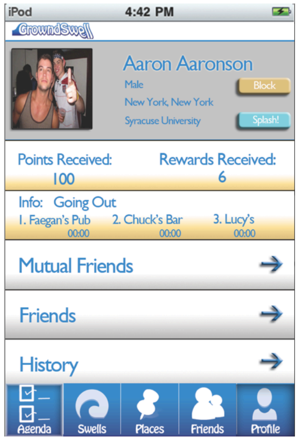
2. **Places_Venue**
    a. Venue Info: Address, City, State and Zip Code. Can we display images specific to the business type (eg. Café, Restaurant, Clothes store etc similar to Foursquare and Gowalla?)
    b. People Here: Button displays a list of people that have currently checked into the venue. Icon displays facebook profile images (eg. Similar to Foursquare). People here (entire tab) will change screen to Places_People_Here. People at the venue might not necessarily be all your friends. If they are your friends you can view their Friend_Profile else you can view a limited info Stranger_Profile. For strangers, you can click on the add button to send a friend request alert to the stranger.
    c. Current Swell offer (including the bar) is only visible if the location currently has a discount. (Not in alpha-version) but code is included to be able to implement this in the future.
    d. People Attending: Button displays a list of people that have added the venue to their agendas. People attending (entire tab) will change screen to Places_People_Attending. Similar to People Here.
    e. Ticker number (in grey box): Displays a list of people who have unlocked the swell. (Eg. 50 means 50 people have received the discount whereas 50/100 means 50 of 100 people have received the discount. This info is retrieved from the database.)
    f. Check-In: User can only check in from close proximity to the venue of about 50 meters max. (Is this possible to implement?) Once the user has checked-in button fades away. Automatically check 30 minutes later if user is at location. User can only check-in to 1 destination so remove user from old venue if user checks-in to a new venue. This adds user on People Here list at the specified venue.
    g. Add to Agenda: Add venue to Going_Out_Summary and fade button. If user removes event from agenda, display button on venue. This adds user on People Attending list at the specified venue.

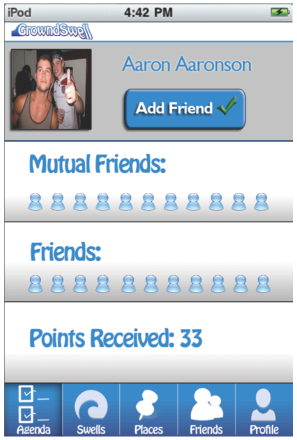
h. Star: Places_Venue page should contain the star that can favortize/unfavortize the venue that then appears in the Places_Favorites.

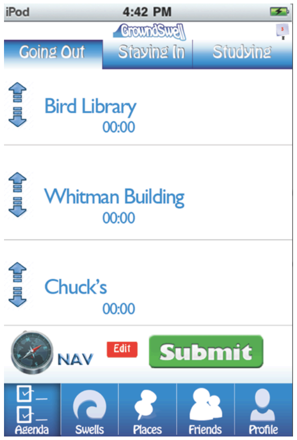


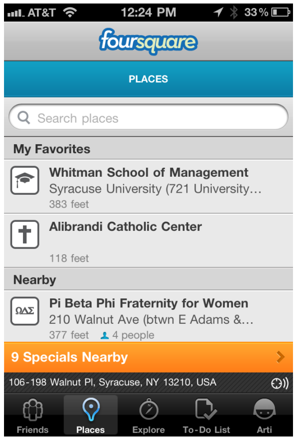
The next 200 people to check-in will receive 15% their first round of drinks.





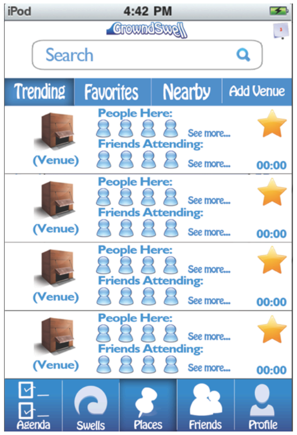


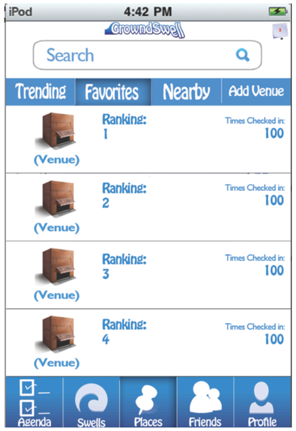


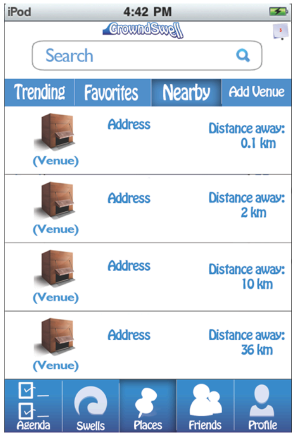

AT&T
12:24 PM
33%
foursquare
PLACES
Search places
My Favorites
Whitman School of Management
Syracuse University (721 University…
383 feet
Alibrandi Catholic Center
118 feet
Nearby
Pi Beta Phi Fraternity for Women
210 Walnut Ave (btwn E Adams &…
377 feet
4 people
9 Specials Nearby
106-198 Walnut Pl, Syracuse, NY 13210, USA
Friends
Places
Explore
To-Do List
Arti

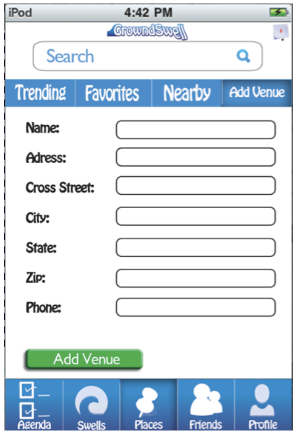
3. **Places_Trending**
    1. Change to ‘People Attending’ and ‘People Here’.
    2. Include a number instead of ‘See more’ or after ‘See more…’ to show the number of entries in the list.
    3. Those people icons are facebook profile pics.
    4. Number on the side is a Male:Female ratio.
    5. Prioritize the trending locations based on ‘People Attending’.
    6. Show ‘People Attending’ on top and ‘People Here’ on bottom.
4. **Places_Favorites**
    1. User can make any venue a favorite by pressing the yellow star on venue page and visa versa.
    2. Red indicates it is a favorite. Yellow indicates it is an unfavorite.
    3. Prioritize order by the number of check-ins displayed on the right hand side.
5. **Places_Nearby**
    1. Displays locations metrics in m.
    2. Search bar is only in Nearby tab.
    3. Last option in table is ‘Location not found’ that redirects cursor to the top to the search bar.
    4. Search bar: If the location searched for is not found, list displays ‘Did you mean?’ and a list similar names places (feature on Google maps) in the table. ‘*Add Venue’* on table list at most bottom of search list that directs user to Places_Add_Venue screen.
6. **Places_Add_Venue**
    1. Enter fields: Venue Name, Address (must be a valid address checked immediately through Google maps else alert user to retry), City, State, Zip, Business/Personal button slide button option, Location Visible To: Friends, Friends of Friends, Public (slide on/off options available). Add venue gets location added to GrowndSwell database and can be removed/edited at the discretion of the company.
    2. Only the ‘Phone’ and ‘Cross Street’ entry is optional.
    3. Include map at bottom to show the pin of the location that the user is entering the location for. Similar to foursquare, it covers half the page.

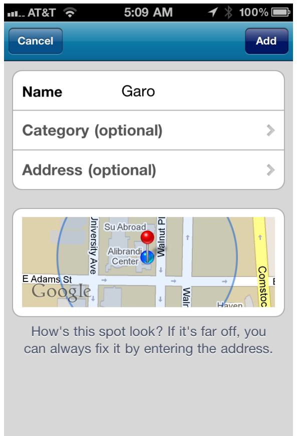
GrowndSwell v1.0 Places Screenshots









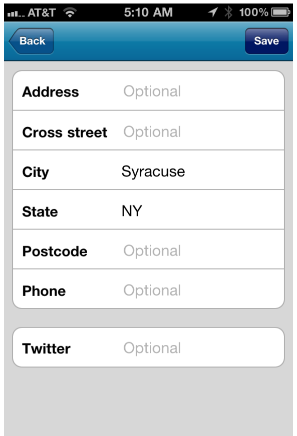


Name Garo

Category (optional)

Address (optional)

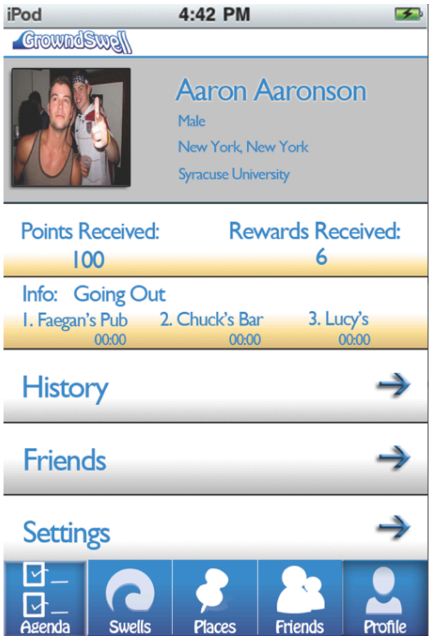
How's this spot look? If it's far off, you can always fix it by entering the address.



Address	Optional
Cross street	Optional
City	Syracuse
State	NY
Postcode	Optional
Phone	Optional
Twitter	Optional

GrowndSwell v1.0 Profile Screenshots

My Profile

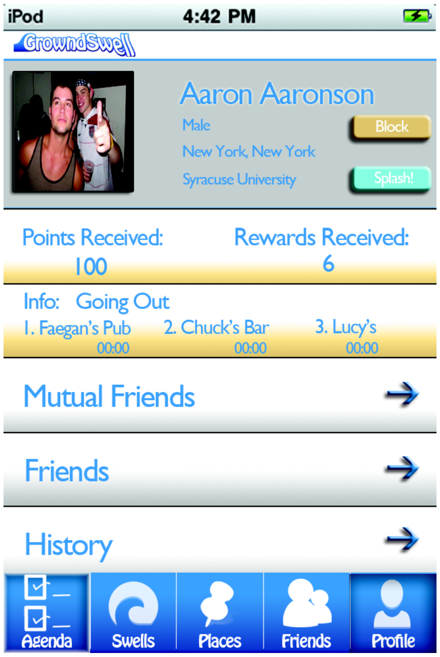


Aaron Aaronson
Male
New York, New York
Syracuse University

Points Received: 100
Rewards Received: 6

Info: Going Out
1. Faegan's Pub 00:00
2. Chuck's Bar 00:00
3. Lucy's 00:00

Friends Profile

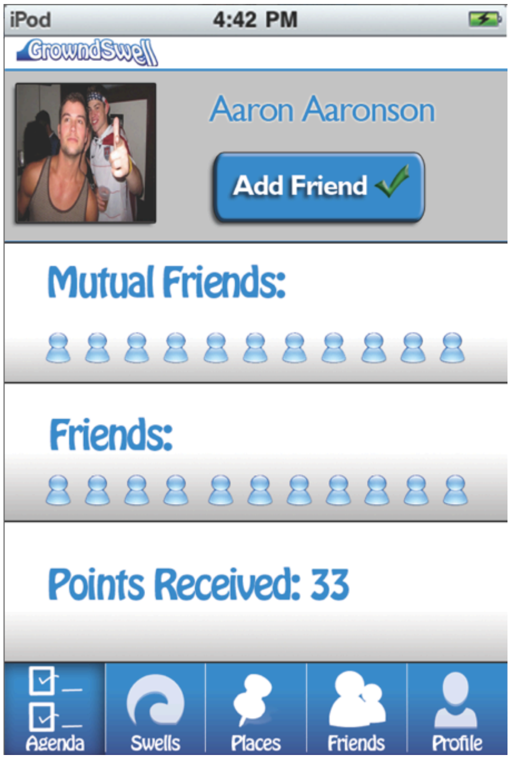


Aaron Aaronson
Male
New York, New York
Syracuse University

Points Received: 100
Rewards Received: 6

Info: Going Out
1. Faegan's Pub 00:00
2. Chuck's Bar 00:00
3. Lucy's 00:00

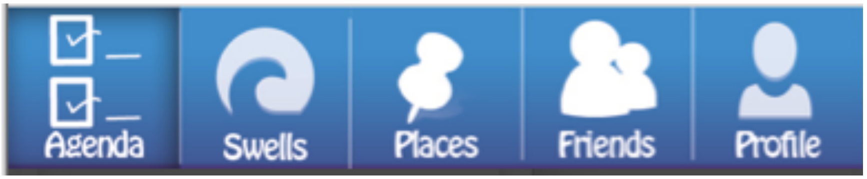
Strangers Profile (Can you view a strangers mutual friends and friends profile details?)



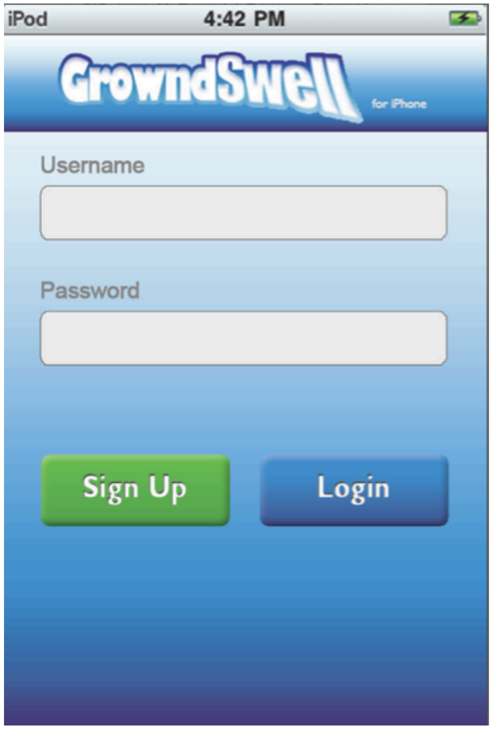
Growndswell

1. **Version 1.1**
   a. Alpha-prototype that is limited to only syr.edu users.
   b. Unlimited installs; account can only be verified as @syr.edu and accepting verification in email.
   c. Version should be compatible with all versions of the iPhone and iTouch
   d. Pulling the view down with finger gives 'Pull down '*Pull down to update…'* and '*Release to update…'* (eg. see Facebook app)

2. **General**
    a. If there is no access to the Internet, display text alert with 'TRY AGAIN' and 'WORK OFFLINE' only once. Try to reconnect every 60 seconds after that without text alerts. Let user be able to view information that is currently on phone in an offline mode such as view friends, the last place checked-in, last swells offered, most recent agenda and complete profile page.
    b. **5 navigation panels** at the bottom of the screen throughout the entirety of the application: Agenda, Swells, Places, Friends, and Profile. Black default color.
    c. Hierarchy of Bars on top:
    Status Bar: Always viewable on iPhone. Black default color.
    Menu Bar: Displays 'Growndswell' Logo in middle with back button where applicable, and Notification icon always aligned to the right.
    Tab Bar: Displays context sensitive tabs that alternate depending on the bottom navigation panel.
    Search Bar: Viewable on selected views including Places (to search for places), Friends (quick way to find friends), etc. Part of table view available in page where friends, shows, movies, games, venues can be searched for.



3. **Login**
    a. First page should include fields such as the Username/Password text fields with the option to Register/Sign-Up.
    b. Username should be filled with grey text displaying ‘example.syr.edu’.
    c. On ‘Login’ alert user if they want to save password on device.
    d. Display ‘Forgot password? We’ll email it to you again’ below the buttons sign-up and login buttons.
    e. Keep user checked in until they Log Out from the profile page.



4. **Registration**

   a. Bars viewable:
      Status bar, Menu bar

   f. Attribute:
      Text Fields:
      First Name, Last Name, Cell (optional), Username, Password, Email (University)
      On/Off buttons:
      Student/Not (No studying feature on application if user is not a student. User is taken directly to Add_Friends after Registration. If a user is a student, then user is directed to course registration page)
      Buttons:
      Male/Female
      Buttons: 'Auto-fill from Address Book' (Fills users info from their phone address book except username/password
      Slide-In Layout on 'Register':
      View Terms of Service, Yes (proceed to Friends_Add_Friends_Page), No (takes user back to registration page)

   g. Cell verification process:
      Send user text message: "Text back 'xxxxx' to register this number with Growndswell".

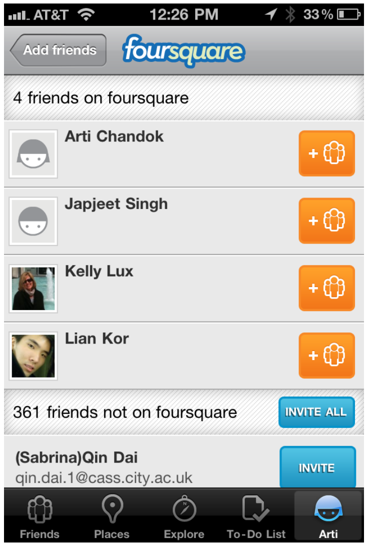
   h. Account verification process: Email user a link (on Syr email currently) that needs to be accepted.

Missing UI: Registration Page
Missing UI: Course Registration Page

5. **Course Registration**
    a. At least 4 courses need to be filled in by the students to be able to continue to the next step.
    b. Add Course button: **Course List** page where students can select from a list created as students fill in courses. Students can select from courses available on database that were populated from other students.
6. **New Course**
    a. Last option on **Course List** page lets students add in their own courses.
    b. Text Fields: Course code ('Enter 3 digit course code'), Course Number ('Enter 3 digit course number'), Professor ('Enter first 3 characters of your professors name').

7. **Friends_Add_Friends**
    a. Add friends through Fb, Twitter, Phone Address book. Use Fb, Twitter API to check if your friend has already downloaded the Growndswell Application, obtain friends email addresses, and have access to post on your Fb, Twitter account.
8. **Facbook_Connect_Layout/Twitter_Connect_Layout**
    a. Overlay views to access information
9. **Friends_Facebook_Add_Friends_Page/ Friends_Address_Book_Add_Friends_Page/ Friends_Twitter_Add_Friends_Page**
    a. 'ADD' and 'INVITE' button fade away when you click.
    b. Friends already added do not show up again.
    c. 'Add' button for friends that have registered.
    d. 'Invite' button for friends that have not registered and use email to send invite.
    e. 'Invite All' button above to invite all friends via email.

Missing UI: Friends List pages for facebook, twitter, address book as shown below. Email messages developed to send pre-filled invitation info.



AT&T 12:26 PM 33 %

Add friends foursquare

4 friends on foursquare

Arti Chandok

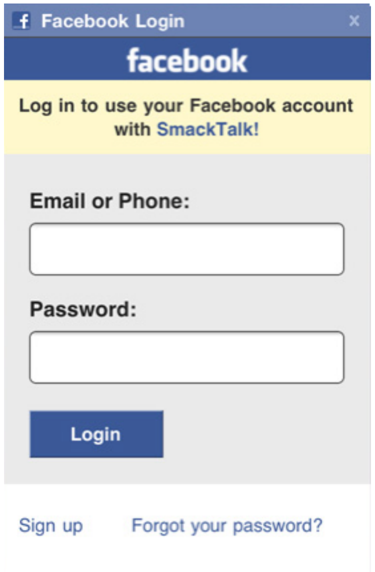
Japjeet Singh

Kelly Lux

Lian Kor

361 friends not on foursquare INVITE ALL

(Sabrina)Qin Dai
qin.dai.1@cass.city.ac.uk INVITE



Facebook Login

facebook

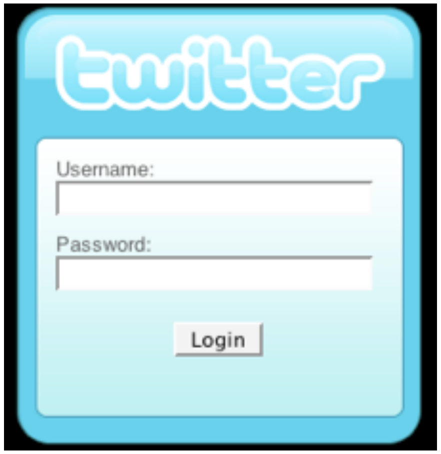
Log in to use your Facebook account with SmackTalk!

Email or Phone:

Password:

Login

Sign up Forgot your password?



twitter

Username:

Password:

Login

10. Profile_Settings (Under Profile Page)

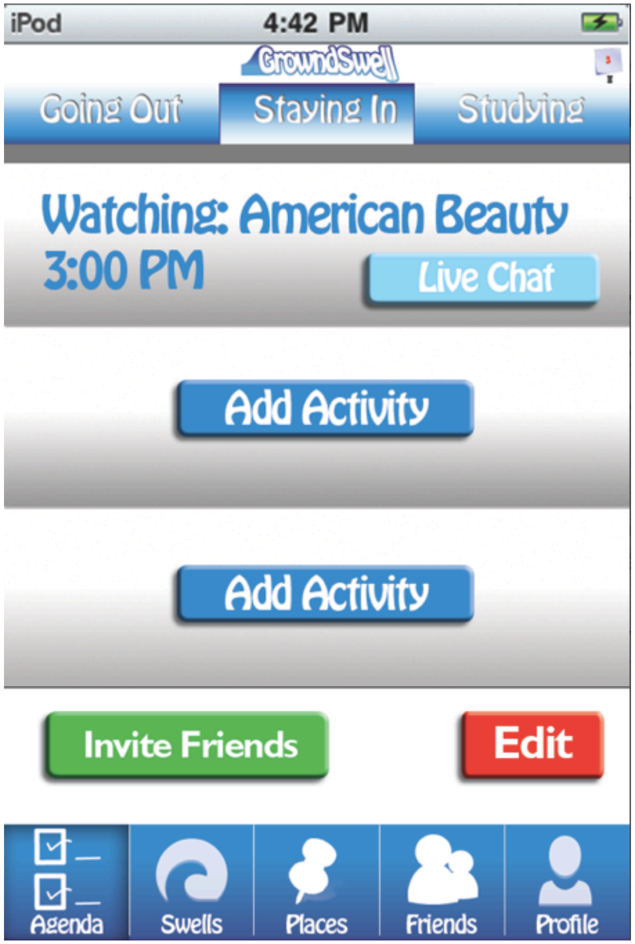
a. Default apple UI page
b. Attributes
   I. Text Fields: (no header UI)
      i. '*Search Nearby Location:*' (User can enter default number for meters. Default is set to 800m)
      ii. Password change (send verification email on email change)
   II. On/Off Switch: (no header UI)
      i. '*Post to Facebook:'*
      ii. 'Post to Twitter:'
      iii. '*Reminder to Event:*' (Default Alert text UI 30 minutes prior to event on agenda)
      iv. Student/Not
   III. Alerts: (header UI included)
      i. Swell Limit Reached (when the swell at the current checked in location is complete)
      ii. Friend Add Agenda Request (when a friend invites you to add their agenda)
      iii. Invitation received (invitation from a stranger to add them as your friend)
      iv. Chat Message received (if someone posts a chat in the chat window where you posted a comment on the show)
   IV. Follow-on Tabs:
      i. 'Friends Check-In' (Can add friends from Growndswell friend list to receive alert if those friends check-in to same place as you)

Most screens are intuitive, however there needs to be some Clarification with some screens:

Agenda_Staying_In_Agenda

Edit: Changes the filled ta to a minus sign so you can remove the inserted activity.
Add Activity: Screen changes to Agenda_Staying_In_Main

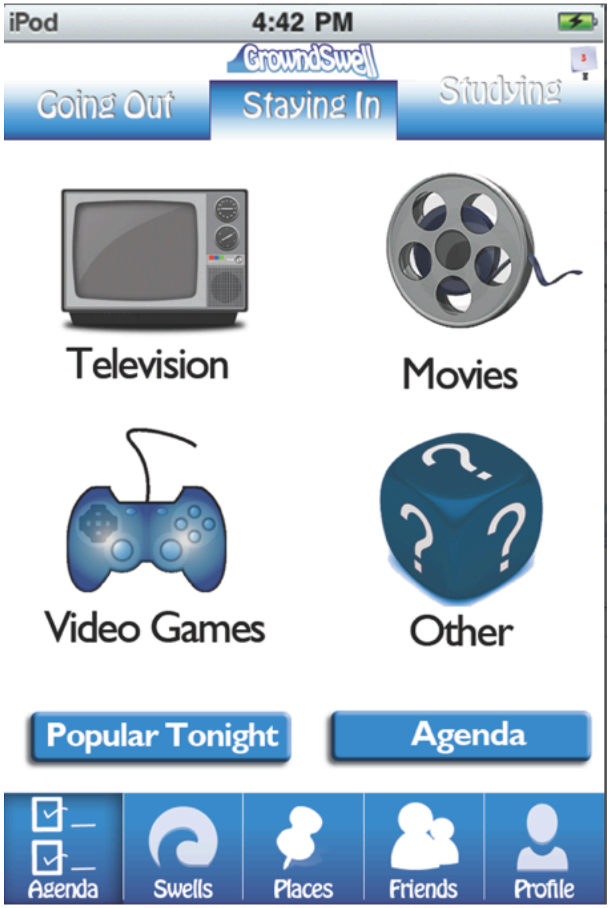
Live Chat: History board that displays a table of chat messages from yesterday, today, 2 days before, 3 days before, 4 days before,…1 week before. Delete all others



Watching: American Beauty
3:00 PM

Agenda_Staying_In_Main

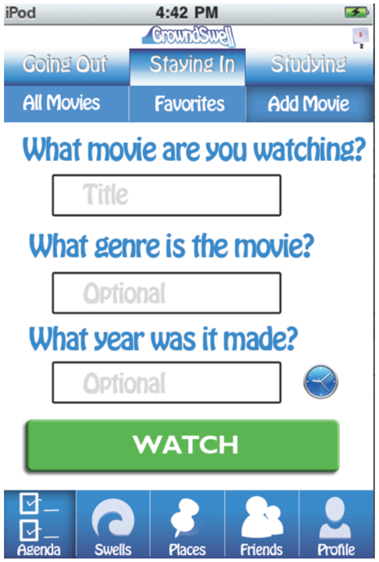
Trending Tonight button (Replaced Trending tonight):
Displays top 5 TV Series, Movies, Video Games tabs that your friends are watching.



GrowndSwell v1.0 Staying_In Screenshots

The tabs have changed to 'Popular Movies', 'Search Movies', and 'Favorite Movies'; 'Popular Shows, 'Search Shows, and 'Favorite Shows'; 'Popular Games, 'Search Games, and 'Favorite Games'

Add Movies, Add Shows, Add Games are a new view that can be accessed through the last row in the 'Search' tab of each category.



What movie are you watching?

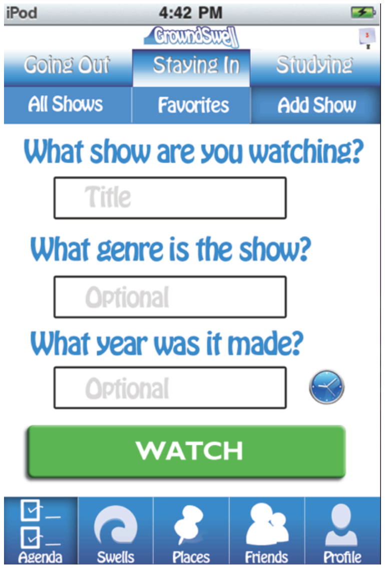
Title

What genre is the movie?

Optional

What year was it made?

Optional



What show are you watching?

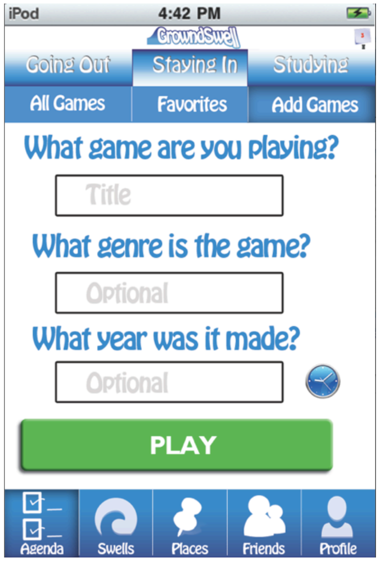
Title

What genre is the show?

Optional

What year was it made?

Optional



What game are you playing?

Title

What genre is the game?

Optional

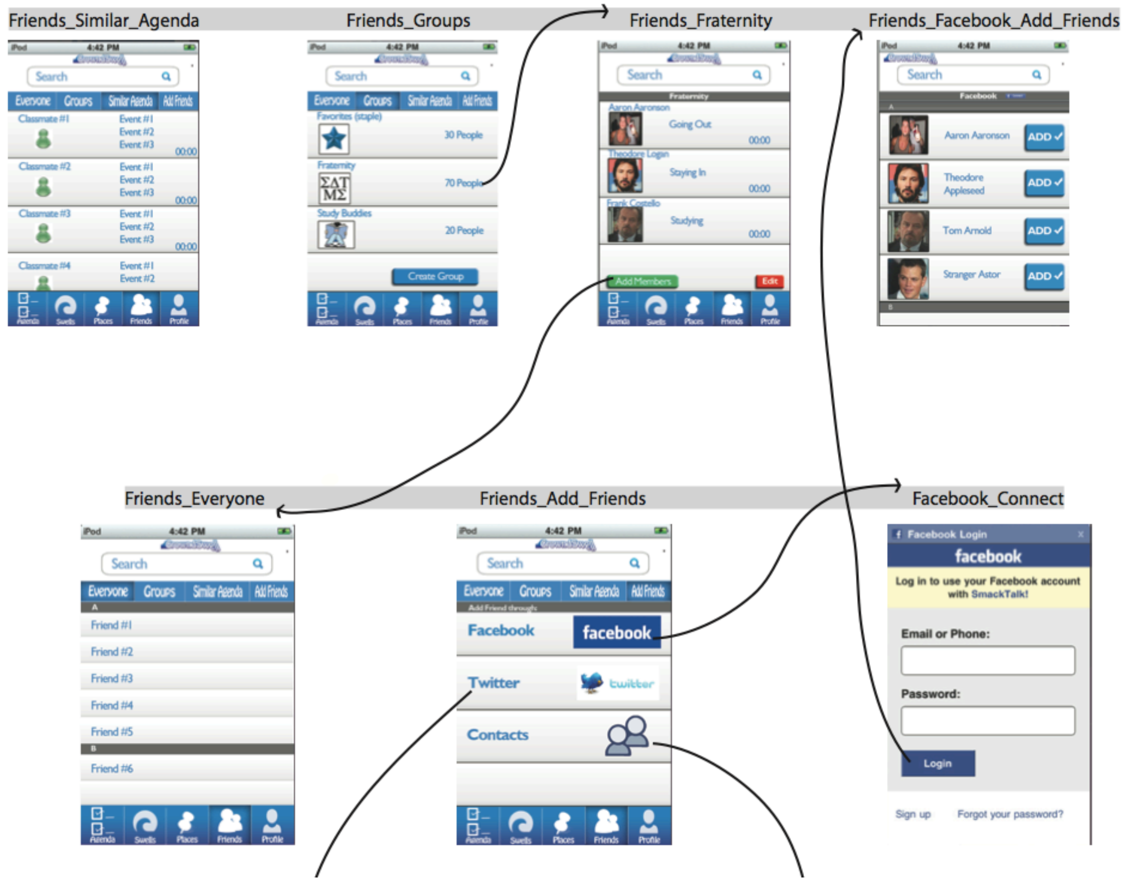
What year was it made?

Optional

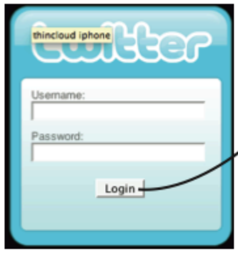
Flowcharts

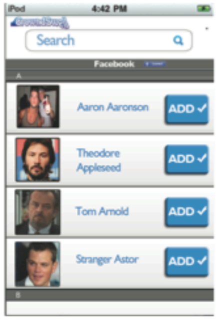
By:

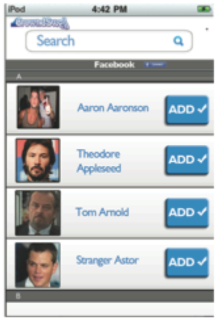
Arti Villa Chandok

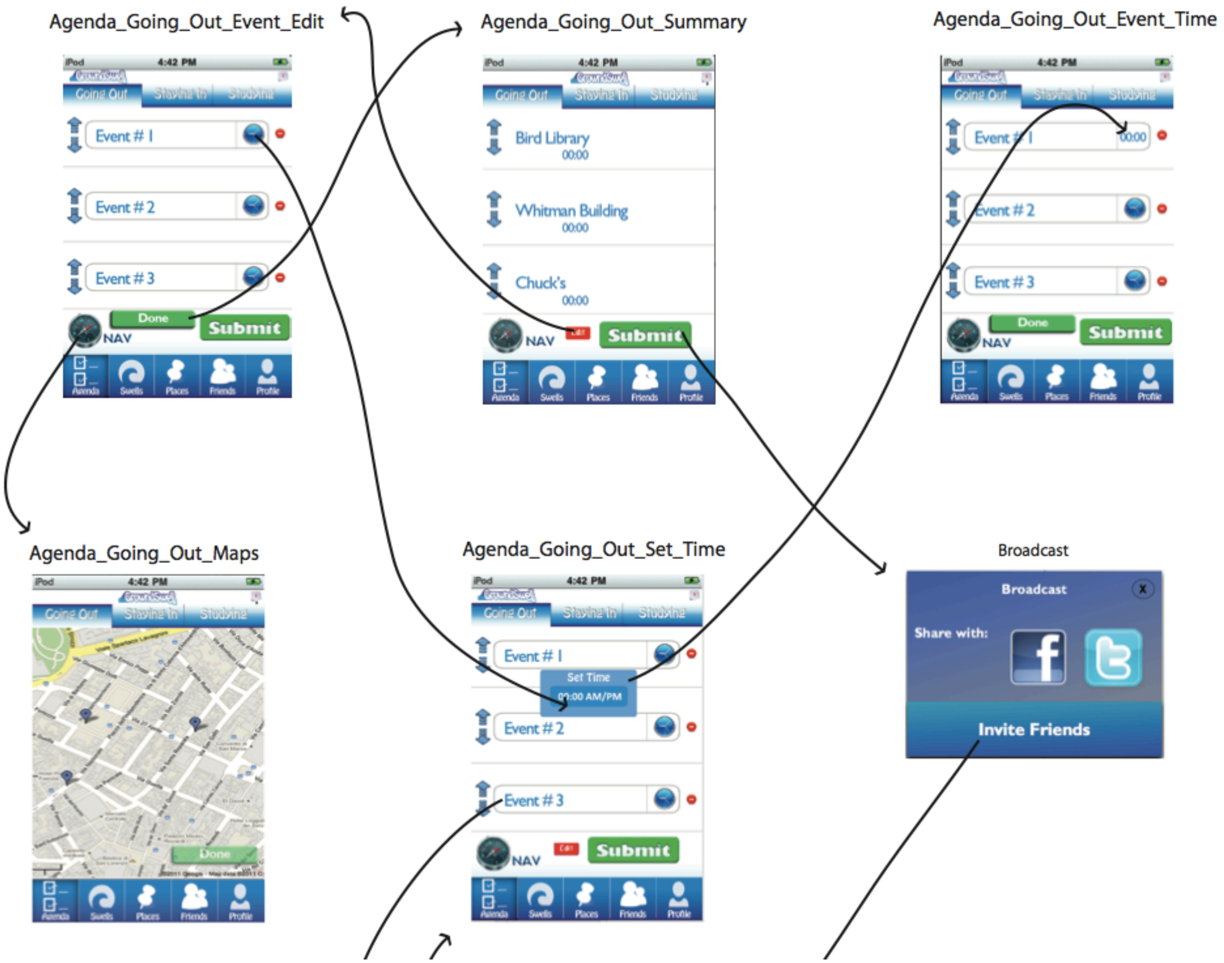

Friends_Similar_Agenda
Friends_Groups
Friends_Fraternity
Friends_Facebook_Add_Friends
Search
Everyone
Groups
Similar Agenda
Add Friends
Classmate #1
Classmate #2
Classmate #3
Classmate #4
Event #1
Event #2
Event #3
00:00
Favorites (staple)
30 People
Fraternity
ΣΑΤ ΜΣ
70 People
Study Buddies
20 People
Create Group
Aaron Aaronson
Going Out
Theodore Logan
Staying In
Frank Costello
Studying
Add Members
Edit
Facebook
Aaron Aaronson
Theodore Appleseed
Tom Arnold
Stranger Astor
ADD ✓
Agenda
Swells
Places
Friends
Profile
Friends_Everyone
Friends_Add_Friends
Facebook_Connect
Friend #1
Friend #2
Friend #3
Friend #4
Friend #5
Friend #6
Add Friend through
Facebook
facebook
Twitter
twitter
Contacts
Facebook Login
facebook
Log in to use your Facebook account with SmackTalk!
Email or Phone:
Password:
Login
Sign up
Forgot your password?

Twitter_Connect Friends_Twitter_Add_Friends Friends_Address_Book_Add_Friends









Agenda_Going_Out_Event_Edit

Agenda_Going_Out_Summary

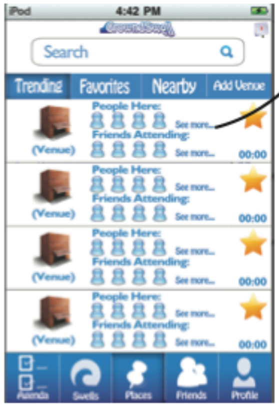
Agenda_Going_Out_Event_Time

Agenda_Going_Out_Maps

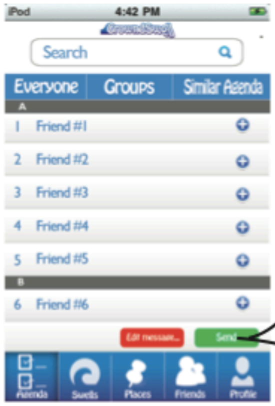
Agenda_Going_Out_Set_Time

Broadcast

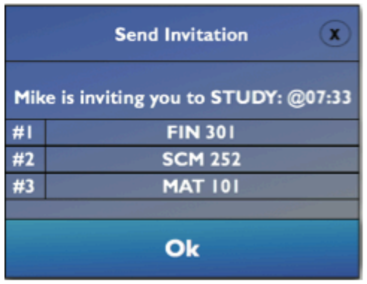
Places_Trending



Friends_Going_Out_Everyone



Send_Invitation



Send Invitation X

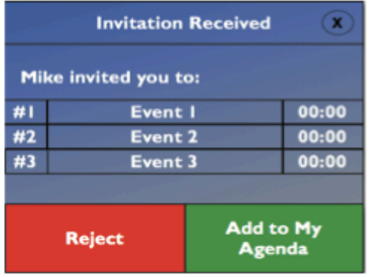
Mike is inviting you to STUDY: @07:33

#1	FIN 301
#2	SCM 252
#3	MAT 101

Ok

(User sending the invitation)

Invitation_Received



Invitation Received X

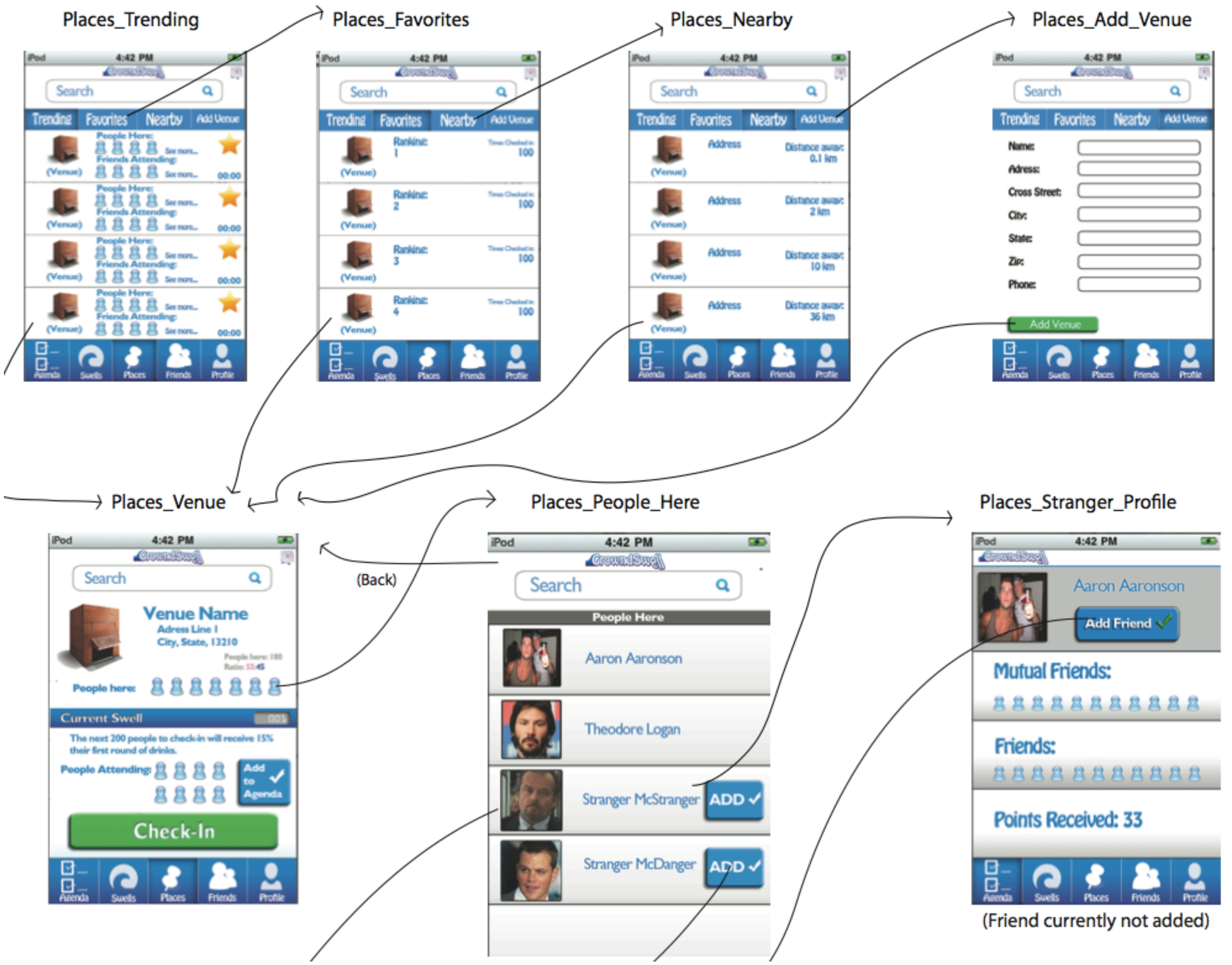
Mike invited you to:

#1	Event 1	00:00
#2	Event 2	00:00
#3	Event 3	00:00

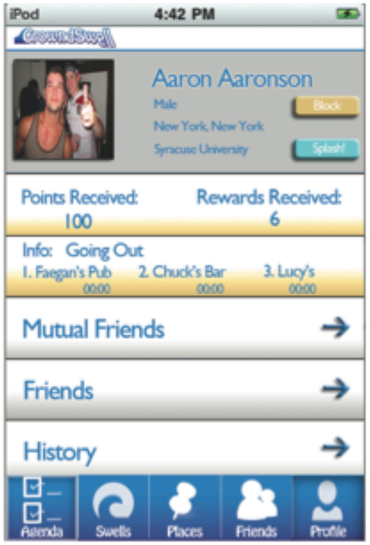
Reject

Add to My Agenda

(User receiving the invitation)


Places_Trending
Places_Favorites
Places_Nearby
Places_Add_Venue
Search
Trending
Favorites
Nearby
Add Venue
People Here:
Friends Attending:
(Venue)
See more...
00:00
Ranking:
1
2
3
4
Times Checked in
100
Address
Distance away:
0.1 km
2 km
10 km
36 km
Name:
Adress:
Cross Street:
City:
State:
Zip:
Phone:
Add Venue
Agenda
Swells
Places
Friends
Profile
Places_Venue
Venue Name
Adress Line 1
City, State, 13210
People here:
Current Swell
The next 200 people to check-in will receive 15% their first round of drinks.
People Attending
Add to Agenda
Check-In
(Back)
Places_People_Here
People Here
Aaron Aaronson
Theodore Logan
Stranger McStranger
ADD ✓
Stranger McDanger
ADD ✓
Places_Stranger_Profile
Aaron Aaronson
Add Friend ✓
Mutual Friends:
Friends:
Points Received: 33
(Friend currently not added)

Friend_Profile ✓



Aaron Aaronson

Male

New York, New York

Syracuse University

Points Received: 100

Rewards Received: 6

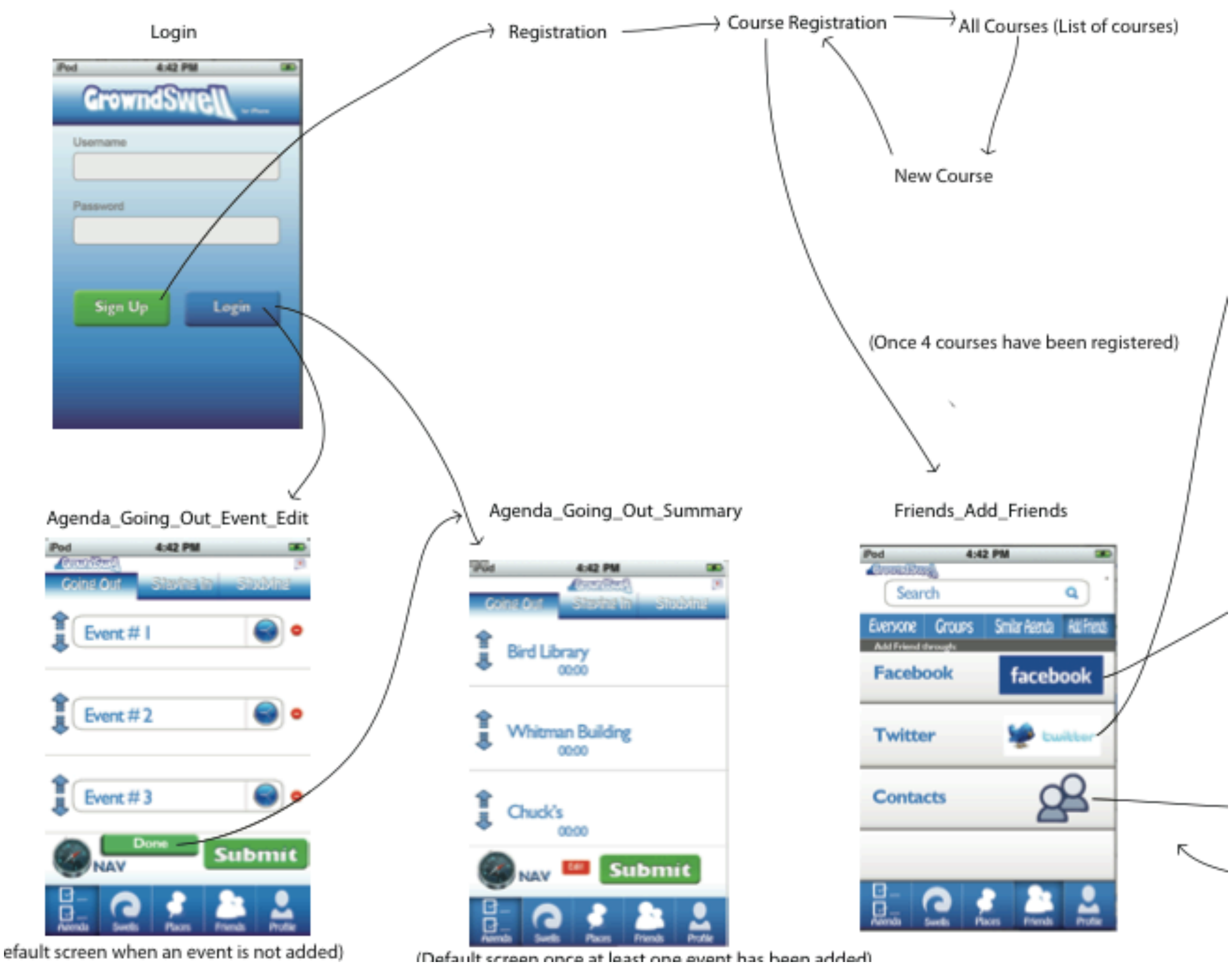
Info: Going Out

1. Faegan's Pub 00:00

2. Chuck's Bar 00:00

3. Lucy's 00:00

(Friend added)



Login

Registration

Course Registration

All Courses (List of courses)

New Course

(Once 4 courses have been registered)

Agenda_Going_Out_Event_Edit

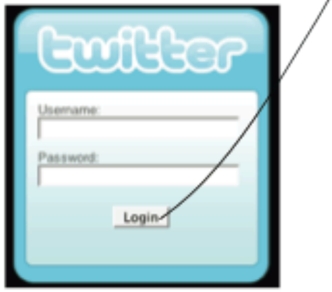
Agenda_Going_Out_Summary

Friends_Add_Friends

efault screen when an event is not added)

(Default screen once at least one event has been added)

Twitter_Connect



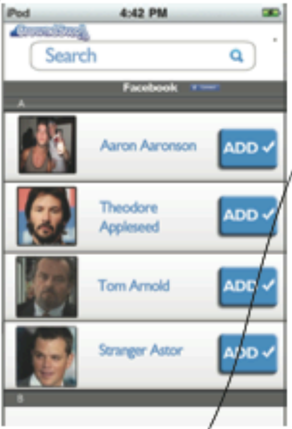
twitter

Username:

Password:

Login

Friends_Twitter_Add_Friends



iPod 4:42 PM

Search

Facebook

A

Aaron Aaronson ADD ✓

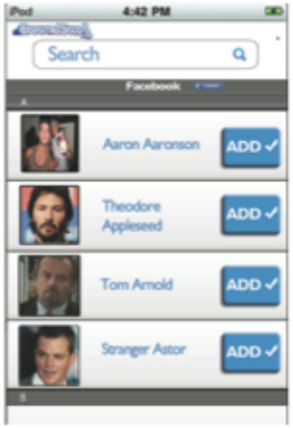
Theodore Appleseed ADD ✓

Tom Arnold ADD ✓

Stranger Astor ADD ✓

B

Friends_Address_Book_Add_Friends



iPod 4:42 PM

Search

Facebook

A

Aaron Aaronson ADD ✓

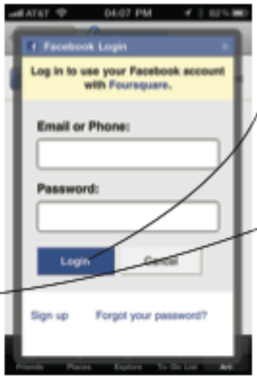
Theodore Appleseed ADD ✓

Tom Arnold ADD ✓

Stranger Astor ADD ✓

B

Facebook_Connect



AT&T 04:07 PM 82%

Facebook Login

Log in to use your Facebook account with Foursquare.

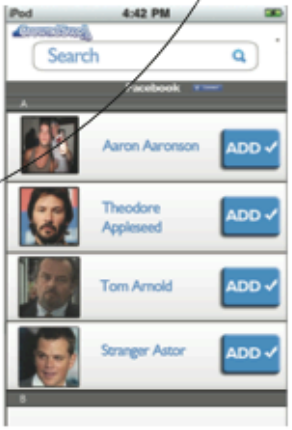
Email or Phone:

Password:

Login Cancel

Sign up Forgot your password?

Friends_Facebook_Add_Friends



iPod 4:42 PM

Search

Facebook

A

Aaron Aaronson ADD ✓

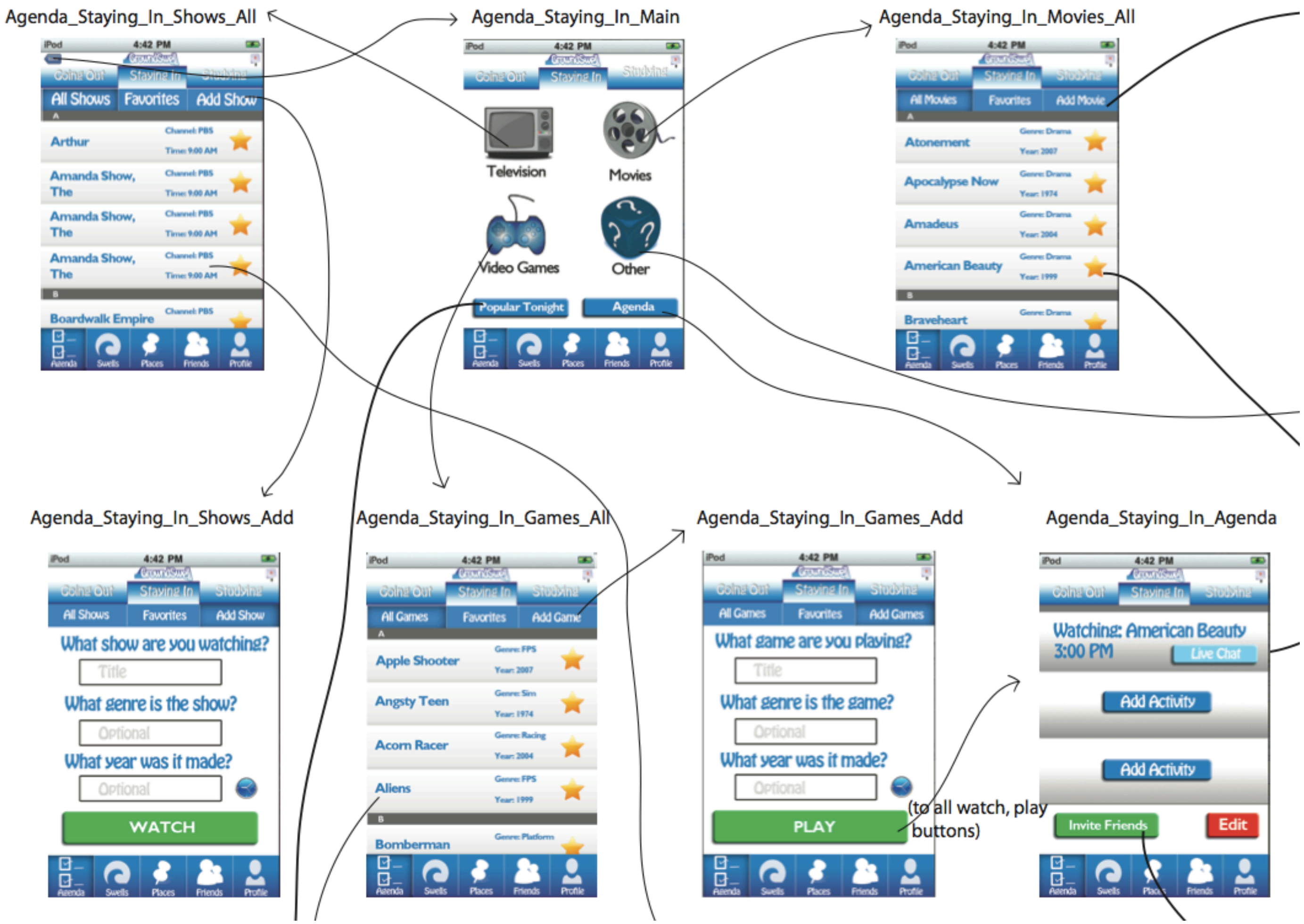
Theodore Appleseed ADD ✓

Tom Arnold ADD ✓

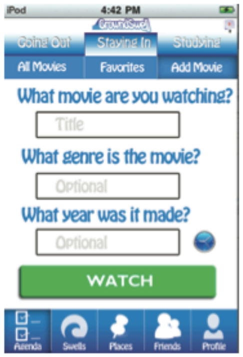
Stranger Astor ADD ✓

B

(Return to Friends_Add_Friends once friends have been added)


Agenda_Staying_In_Shows_All
Agenda_Staying_In_Main
Agenda_Staying_In_Movies_All
Agenda_Staying_In_Shows_Add
Agenda_Staying_In_Games_All
Agenda_Staying_In_Games_Add
Agenda_Staying_In_Agenda
iPod
4:42 PM
Going Out
Staying In
Studying
All Shows
Favorites
Add Show
Arthur
Amanda Show, The
Boardwalk Empire
Channel: PBS
Time: 9:00 AM
Television
Movies
Video Games
Other
Popular Tonight
Agenda
All Movies
Add Movie
Atonement
Apocalypse Now
Amadeus
American Beauty
Braveheart
Genre: Drama
What show are you watching?
Title
What genre is the show?
Optional
What year was it made?
WATCH
All Games
Add Game
Apple Shooter
Angsty Teen
Acorn Racer
Aliens
Bomberman
Add Games
What game are you playing?
What genre is the game?
PLAY
(to all watch, play buttons)
Watching: American Beauty
3:00 PM
Live Chat
Add Activity
Add Activity
Invite Friends
Edit
Agenda
Swells
Places
Friends
Profile

Agenda_Staying_In_Movies_Add



What movie are you watching?

Title

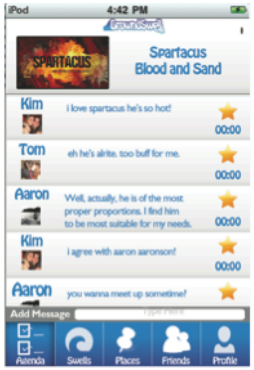
What genre is the movie?

Optional

What year was it made?

Optional

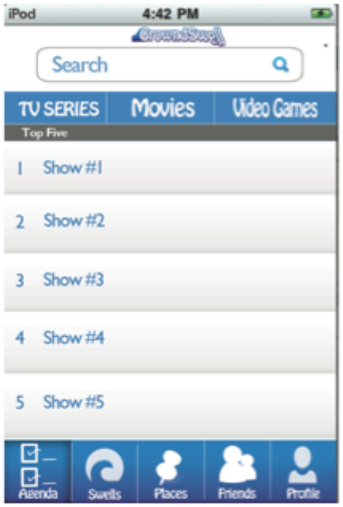
Agenda_Staying_In_Shows_Chat



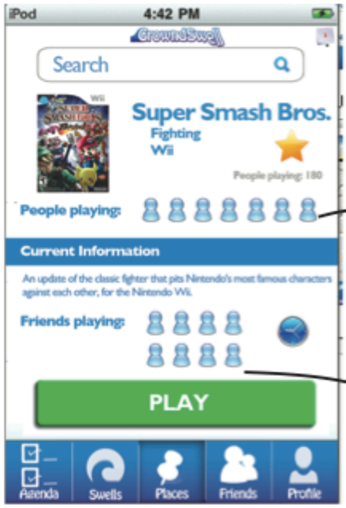
Spartacus
Blood and Sand

Kim	i love spartacus he's so hot!	00:00
Tom	eh he's alrite. too buff for me.	00:00
Aaron	Well, actually, he is of the most proper proportions. I find him to be most suitable for my needs.	00:00
Kim	i agree with aaron aaronson!	00:00
Aaron	you wanna meet up sometime?	

Agenda_Staying_In_Popular

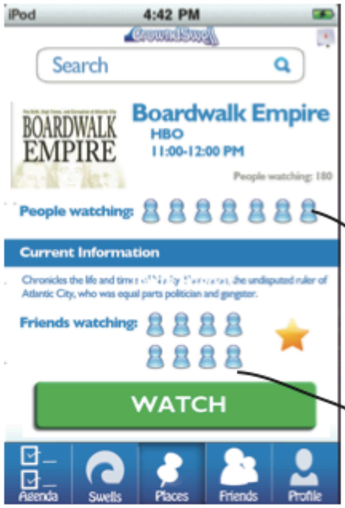


Agenda_Staying_In_Games_Profile



An update of the classic fighter that pits Nintendo's most famous characters against each other, for the Nintendo Wii.

Agenda_Staying_In_Shows_Profile



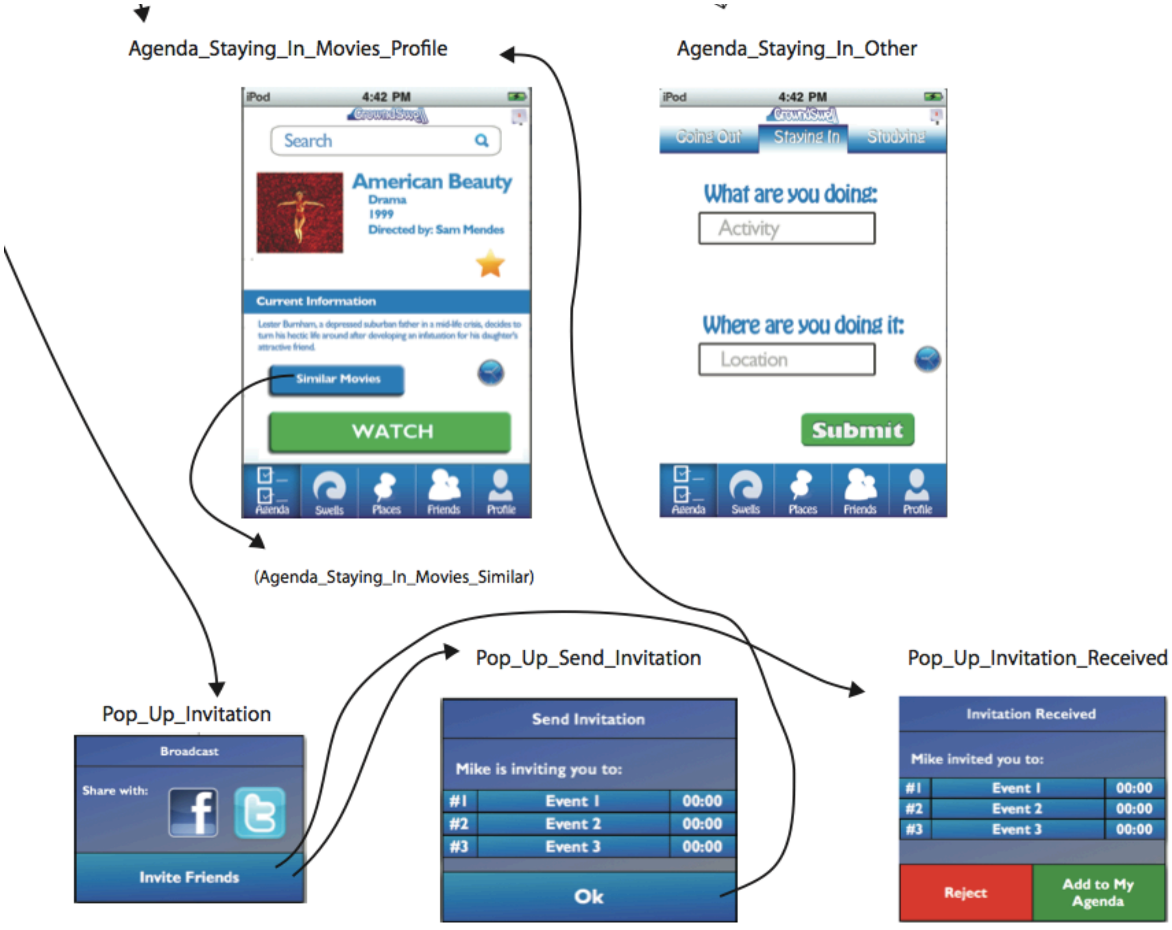
Chronicles the life and times [illegible], the undisputed ruler of Atlantic City, who was equal parts politician and gangster.

(Agenda_Staying_In_Games_People_Playing)

(Agenda_Staying_In_Games_Friends_Playing)

(Agenda_Staying_In_Shows_People_Watching)

(Agenda_Staying_In_Shows_Friends_Watching)



Agenda_Staying_In_Movies_Profile

American Beauty

Lester Burnham, a depressed suburban father in a mid-life crisis, decides to turn his hectic life around after developing an infatuation for his daughter's attractive friend.

(Agenda_Staying_In_Movies_Similar)

Agenda_Staying_In_Other

What are you doing:

Activity

Where are you doing it:

Location

Pop_Up_Invitation

Share with:

Pop_Up_Send_Invitation

Mike is inviting you to:

#1	Event 1	00:00
#2	Event 2	00:00
#3	Event 3	00:00

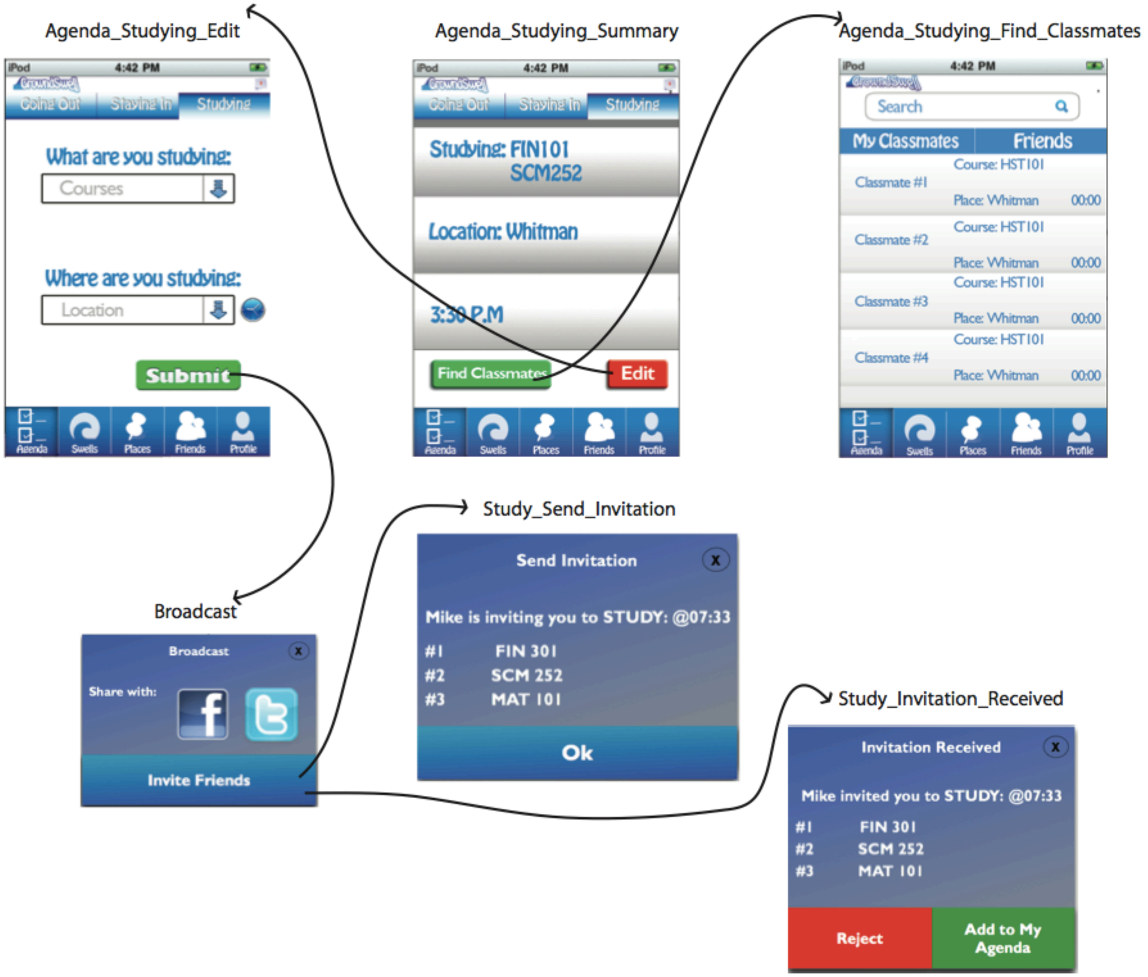
(User sending the invitation)

Pop_Up_Invitation_Received

Mike invited you to:

#1	Event 1	00:00
#2	Event 2	00:00
#3	Event 3	00:00

(User receiving the invitation)



What are you studying:

Where are you studying:

Studying: FIN101 SCM252

Location: Whitman

3:30 P.M

Classmate #1	Course: HST101 Place: Whitman	00:00
Classmate #2	Course: HST101 Place: Whitman	00:00
Classmate #3	Course: HST101 Place: Whitman	00:00
Classmate #4	Course: HST101 Place: Whitman	00:00

Mike is inviting you to STUDY: @07:33

#1 FIN 301
#2 SCM 252
#3 MAT 101

Mike invited you to STUDY: @07:33

#1 FIN 301
#2 SCM 252
#3 MAT 101

Source Code

By:

Arti Villa Chandok

```
//
//  AgendaStayingInViewController.h
//  GrowndSwell
//
//  Created by ArtiVilla on 4/3/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import <UIKit/UIKit.h>


@interface AgendaStayingInViewController : UIViewController
{

    IBOutlet UIImageView *topBar;
    IBOutlet UILabel *activitName;
    IBOutlet UILabel *activitName1;
    IBOutlet UILabel *activitName2;

    IBOutlet UIButton *activiyButton;
    IBOutlet UIButton *activiyButton1;
    IBOutlet UIButton *activiyButton2;

}

@property (nonatomic,retain) IBOutlet UILabel *activitName;
@property (nonatomic,retain) IBOutlet UILabel
*activitName1;
@property (nonatomic,retain) IBOutlet UILabel
*activitName2;
@property (nonatomic,retain) IBOutlet UIButton
*activitButton;
@property (nonatomic,retain) IBOutlet UIButton
*activitButton1;
@property (nonatomic,retain) IBOutlet UIButton
*activitButton2;


//Method
-(IBAction)addActivity:(id)sender;
-(IBAction)liveChat:(id)sender;

@end
```

```
//
//  AgendaStayingInViewController.m
//  GrowndSwell
//
//  Created by ArtiVilla on 4/3/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import "AgendaStayingInViewController.h"


@implementation AgendaStayingInViewController

@synthesize
activitName,activitName1,activitName2,activitButton,activit
Button1,activitButton2;

// The designated initializer.  Override if you create the
controller programmatically and want to perform
customization that is not appropriate for viewDidLoad.
/*
- (id)initWithNibName:(NSString *)nibNameOrNil
bundle:(NSBundle *)nibBundleOrNil {
    self = [super initWithNibName:nibNameOrNil
bundle:nibBundleOrNil];
    if (self) {
        // Custom initialization.
    }
    return self;
}
*/

-(IBAction)addActivity:(id)sender
{

}

-(IBAction)liveChat:(id)sender
{
}
/*
// Implement viewDidLoad to do additional setup after
loading the view, typically from a nib.
- (void)viewDidLoad {
    [super viewDidLoad];
}
*/

/*
// Override to allow orientations other than the default
```

```
portrait orientation.
-
(BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOr
ientation)interfaceOrientation {
    // Return YES for supported orientations.
    return (interfaceOrientation ==
UIInterfaceOrientationPortrait);
}
*/

- (void)didReceiveMemoryWarning {
    // Releases the view if it doesn't have a superview.
    [super didReceiveMemoryWarning];

    // Release any cached data, images, etc. that aren't in
use.
}

- (void)viewDidUnload {
    [super viewDidUnload];
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}


- (void)dealloc {
    [super dealloc];
}


@end
```

```
//
//  AgendaViewController.h
//  GrowndSwell
//
//  Created by ArtiVilla on 3/18/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import <UIKit/UIKit.h>
@interface AgendaViewController : UIViewController {

    IBOutlet UIImageView *topBannerView;
    IBOutlet UIImageView *studAreaBack;
    IBOutlet UIImageView *locationArea;
    IBOutlet UIImageView *timeArea;
    IBOutlet UIButton *goingOutButton;
    IBOutlet UIButton *studyingOutButton;
    IBOutlet UIButton *studyingButton;
    IBOutlet UIButton *navButton;
    IBOutlet UIButton *editButton;
    IBOutlet UIButton *submitButton;

    IBOutlet UIButton *subButton;


    ///Study method

    IBOutlet UIImageView *rowBack1;
    IBOutlet UIImageView *rowBack2;
    IBOutlet UIImageView *rowBack3;
    IBOutlet UIView *studyBackView;
    IBOutlet UILabel *studyLable;
    IBOutlet UILabel *studyLableValue;
    IBOutlet UILabel *studyLable1Value;
    IBOutlet UILabel *locationLabel;
    IBOutlet UILabel *locationLabelValue;
    IBOutlet UILabel *timeLabelValue;
    IBOutlet UIButton *findClassMateButton;
    IBOutlet UIButton *editStudyButton;

    //Going out controller's
    IBOutlet    UIView *goingOutView;
    IBOutlet    UIView *rowOneView;
    IBOutlet    UIView *rowTwoView;
    IBOutlet    UIView *rowThreeView;
    IBOutlet UIButton *upArrowButton;
    IBOutlet UIButton *downArrayButton;
    IBOutlet UIButton *upArrowButton1;
    IBOutlet UIButton *downArrayButton1;
    IBOutlet UIButton *upArrowButton2;
    IBOutlet UIButton *downArrayButton2;
    IBOutlet UILabel *eventLable;
    IBOutlet UILabel *timeLabel;
    IBOutlet UILabel *eventLable1;
    IBOutlet UILabel *timeLabel1;
    IBOutlet UILabel *eventLable2;
    IBOutlet UILabel *timeLabel2;

    //StudyingView Controller
    IBOutlet    UIView *studyingView;
    IBOutlet    UIButton *tvView;
    IBOutlet    UIButton *moviesView;
    IBOutlet    UIButton *vgView;
    IBOutlet    UIButton *otherView;
    IBOutlet    UIButton *popularTonightButton;
    IBOutlet    UIButton *agendaButton;


}






//Studying controller
@property (nonatomic,retain) IBOutlet UIView *studyingView;
@property (nonatomic,retain) IBOutlet UIButton *tvView;
@property (nonatomic,retain) IBOutlet UIButton *moviesView;
@property (nonatomic,retain) IBOutlet UIButton *vgView;
@property (nonatomic,retain) IBOutlet UIButton *otherView;
@property (nonatomic,retain) IBOutlet UIButton
*popularTonightButton;
@property (nonatomic,retain) IBOutlet UIButton
*agendaButton;

//Going out controller
@property (nonatomic,retain) IBOutlet UIButton
*upArrowButton;
@property (nonatomic,retain) IBOutlet UIButton
*downArrayButton;
@property (nonatomic,retain) IBOutlet UIButton
*upArrowButton1;
@property (nonatomic,retain) IBOutlet UIButton
*downArrayButton1;
@property (nonatomic,retain) IBOutlet UIButton
*upArrowButton2;
@property (nonatomic,retain) IBOutlet UIButton
*downArrayButton2;
@property (nonatomic,retain) IBOutlet UIView *goingOutView;
```

```
@property (nonatomic,retain) IBOutlet UIView *rowOneView;
@property (nonatomic,retain) IBOutlet UIView *rowTwoView;
@property (nonatomic,retain) IBOutlet UIView *rowThreeView;
@property (nonatomic,retain) IBOutlet UILabel *eventLable;
@property (nonatomic,retain) IBOutlet UILabel *timeLabel;
@property (nonatomic,retain) IBOutlet UILabel *eventLable1;
@property (nonatomic,retain) IBOutlet UILabel *timeLabel1;
@property (nonatomic,retain) IBOutlet UILabel *eventLable2;
@property (nonatomic,retain) IBOutlet UILabel *timeLabel2;


@property (nonatomic,retain) IBOutlet UIButton
*editStudyButton;
@property (nonatomic,retain) IBOutlet UILabel *courseLabel;
@property (nonatomic,retain) IBOutlet UILabel
*locationLabel;
@property (nonatomic,retain) IBOutlet UILabel
*timeLabelValue;
@property (nonatomic,retain) IBOutlet UITextField
*courseTextField;
@property (nonatomic,retain) IBOutlet UITextField
*locationTextField;
@property (nonatomic,retain) IBOutlet UIButton
*findClassMateButton;
@property (nonatomic,retain) IBOutlet UIButton *subButton;
@property (nonatomic,retain) IBOutlet UIImageView
*topBannerView;
@property (nonatomic,retain) IBOutlet UIImageView
*studAreaBack;
@property (nonatomic,retain) IBOutlet UIImageView
*locationArea;
@property (nonatomic,retain) IBOutlet UIImageView
*timeArea;
@property (nonatomic,retain) IBOutlet UIButton
*goingOutButton;
@property (nonatomic,retain) IBOutlet UIButton
*studyingOutButton;
@property (nonatomic,retain) IBOutlet UIButton
*studyingButton;
@property (nonatomic,retain) IBOutlet UIButton *navButton;
@property (nonatomic,retain) IBOutlet UIButton *editButton;
@property (nonatomic,retain) IBOutlet UIButton
*submitButton;
@property (nonatomic,retain) IBOutlet UITableView
*agendaTable;

//Study View Variables
@property (nonatomic,retain) IBOutlet UIView
*studyBackView;
@property (nonatomic,retain) IBOutlet UIImageView
```

```
*rowBack1;
@property (nonatomic,retain) IBOutlet UIImageView
*rowBack2;
@property (nonatomic,retain) IBOutlet UIImageView
*rowBack3;
@property (nonatomic,retain) IBOutlet UILabel *studyLable;
@property (nonatomic,retain) IBOutlet UILabel
*studyLableValue;
@property (nonatomic,retain) IBOutlet UILabel
*studyLable1Value;
@property (nonatomic,retain) IBOutlet UILabel
*locationLabelValue;

//Instance methods
-(IBAction)editMethod:(id)sender;
-(IBAction)navMethod:(id)sender;
-(IBAction)submitMethod:(id)sender;
-(IBAction)goingOutMethod:(id)sender;
-(IBAction)studyingInMethod:(id)sender;
-(IBAction)studyingMethod:(id)sender;
-(IBAction)subCourseDetails:(id)sender;
-(void)ShowStudying:(BOOL)value;
-(IBAction)findClassMateMethod:(id)sender;
-(IBAction)editStudying:(id)sender;
-(IBAction)movingButton:(id)sender;
-(void)ShowStudying:(BOOL)value;
-(void)eventDetails;
-(void)showGoingOut:(BOOL)value;
-(IBAction)populatTonight:(BOOL)value;
-(IBAction)agendaButtonMethod:(BOOL)value;
-(IBAction)studyingInEvents:(id)sender;


@end
```

```
//
//  AgendaViewController.m
//  GrowndSwell
//
//  Created by ArtiVilla on 3/18/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import "AgendaViewController.h"
#import "GrowndSwellAppDelegate.h"
#import "EditStudyingController.h"
#import "ClassMateClass.h"
#import "PeopleHereViewController.h"
#import "MapViewController.h"
#import "AgendaStayingInViewController.h"
#import "StudyingAllViewController.h"
#import "PopularViewController.h"

@implementation AgendaViewController

@synthesize
topBannerView,studyingOutButton,studyingButton,goingOutButt
on,navButton,editButton,submitButton,agendaTable;
@synthesize subButton,studyingView;
@synthesize locationTextField,courseTextField;
@synthesize courseLabel,locationLabel;
@synthesize
studAreaBack,locationArea,timeArea,editStudyButton,findClas
sMateButton;
@synthesize locationLabelValue,timeLabelValue;
@synthesize
goingOutView,rowOneView,rowTwoView,rowThreeView,upArrowButt
on,downArrayButton,upArrowButton1,downArrayButton1,upArrowB
utton2,downArrayButton2;//Going out Controller
@synthesize
eventLable,timeLabel,eventLable1,timeLabel1,eventLable2,tim
eLabel2;
@synthesize
agendaButton,popularTonightButton,tvView,moviesView,vgView,
otherView;
@synthesize
studyBackView,rowBack1,rowBack2,rowBack3,studyLable,studyLa
bleValue,studyLable1Value;

-(id) initWithTabBar
{
    self = [super init];
    if (self)
    {
     //this is the label on the tab button itself
```

```
        self.title = @"Agenda";
        //parsingType = [[NSString 
alloc]initWithString:@"recent"];
        //Image of the Tab
        //loadStatus = @"loading";
        self.tabBarItem.image = [UIImage 
imageNamed:@"featured.png"];
    }
    return self;
}

-(void)viewWillAppear:(BOOL)animated
{
    self.navigationController.navigationBarHidden = YES;
    self.agendaTable.editing = YES;
}

// The designated initializer.  Override if you create the 
controller programmatically and want to perform 
customization that is not appropriate for viewDidLoad.
/*
- (id)initWithNibName:(NSString *)nibNameOrNil 
bundle:(NSBundle *)nibBundleOrNil {
    self = [super initWithNibName:nibNameOrNil 
bundle:nibBundleOrNil];
    if (self) {
        // Custom initialization.
    }
    return self;
}
*/

-(IBAction)editStudying:(id)sender
{
    EditStudyingController *EditStudyingControllerObj = 
[[EditStudyingController alloc] init];
    [self.navigationController 
pushViewController:EditStudyingControllerObj animated:YES];
    [EditStudyingControllerObj release];

}
-(IBAction)editMethod:(id)sender
{
}

-(IBAction)findClassMateMethod:(id)sender
{
    PeopleHereViewController *ClassMateClassObj = 
[[PeopleHereViewController alloc] init];
    [self.navigationController 
```

```
pushViewController:ClassMateClassObj animated:YES];
    [ClassMateClassObj release];
}
-(IBAction)navMethod:(id)sender
{
    MapViewController *MapViewControllerObj = 
[[MapViewController alloc] init];
    [[self navigationController] 
pushViewController:MapViewControllerObj animated:YES];
    [MapViewControllerObj release];

}

//-(IBAction)hideMap:(id)sender
//{
//  self.mapView.hidden = YES;
//  self.hideMap.hidden = YES;
//}

-(IBAction)submitMethod:(id)sender
{
    NSLog(@"submit button");
    GrowndSwellAppDelegate *GrowndSwellAppDelegateObj = 
(GrowndSwellAppDelegate*)[[UIApplication sharedApplication] 
delegate];
    GrowndSwellAppDelegateObj.bottomTabBar.selectedIndex = 
3;
}
-(IBAction)goingOutMethod:(id)sender
{
    [self ShowStudying:YES];
    [self ShowStudyingIn:YES];
    [self showGoingOut:NO];

}
-(IBAction)studyingInMethod:(id)sender
{
    [self ShowStudying:YES];
    [self ShowStudyingIn:NO];
    [self showGoingOut:YES];

}
-(IBAction)studyingMethod:(id)sender
{
    [self ShowStudying:NO];
    [self ShowStudyingIn:YES];
    [self showGoingOut:YES];
}
```

```
-(void)touchesBegan:(NSSet *)touches withEvent:(UIEvent
*)event
{
    NSUInteger numTaps = [[touches anyObject] tapCount];
    UITouch *touchView = [touches anyObject];

    if(touchView.view == self.eventLable)
    {
     [[NSUserDefaults standardUserDefaults] setValue:@""
forKey:@"eventName"];
     [self eventDetails];
    }
    else if(touchView.view == self.eventLable1)
    {
     [[NSUserDefaults standardUserDefaults] setValue:@""
forKey:@"eventName"];
     [self eventDetails];
    }
    else if(touchView.view == self.eventLable2)
    {
     [[NSUserDefaults standardUserDefaults] setValue:@""
forKey:@"eventName"];
     [self eventDetails];
    }
    /*else if(touchView.view == self.tvView)
    {
     [[NSUserDefaults standardUserDefaults] setValue:@""
forKey:@"eventName"];
     [self studyingInEvents];
    }
    else if(touchView.view == self.moviesView)
    {
     [[NSUserDefaults standardUserDefaults] setValue:@""
forKey:@"eventName"];
     [self studyingInEvents];
    }
    else if(touchView.view == self.vgView)
    {
     [[NSUserDefaults standardUserDefaults] setValue:@""
forKey:@"eventName"];
     [self studyingInEvents];
    }
    else if(touchView.view == self.otherView)
    {
     [[NSUserDefaults standardUserDefaults] setValue:@""
forKey:@"eventName"];
     [self studyingInEvents];
    }
    */
```

```
}

-(void)eventDetails
{
    NSLog(@"submit button");
    GrowndSwellAppDelegate *GrowndSwellAppDelegateObj =
(GrowndSwellAppDelegate*)[[UIApplication sharedApplication]
delegate];
    GrowndSwellAppDelegateObj.bottomTabBar.selectedIndex =
2;

}

-(IBAction)studyingInEvents:(id)sender
{
    StudyingAllViewController *StudyingAllViewControllerObj
= [[StudyingAllViewController alloc]
initWithNibName:@"StudyingAllViewController" bundle:nil];
    [[self navigationController]
pushViewController:StudyingAllViewControllerObj
animated:YES];
    [StudyingAllViewControllerObj release];
}


-(void)ShowStudying:(BOOL)value
{
    self.studyBackView.hidden = value;
    self.rowBack1.hidden = value;
    self.rowBack2.hidden = value;
    self.rowBack3.hidden = value;
    self.locationLabel.hidden = value;
    self.locationLabelValue.hidden = value;
    self.editStudyButton.hidden = value;
    self.findClassMateButton.hidden = value;
    self.studyLable.hidden = value;
    self.studyLable1Value.hidden = value;
    self.studyLableValue.hidden = value;
    self.locationLabelValue.hidden = value;
    self.locationLabel.hidden = value;
    self.timeLabelValue.hidden = value;
}

-(void)showGoingOut:(BOOL)value
{
    self.goingOutView.hidden = value;
    self.rowOneView.hidden = value;
    self.rowOneView.hidden = value;
    self.rowTwoView.hidden = value;
    self.rowThreeView.hidden = value;
    self.upArrowButton.hidden = value;
    self.downArrayButton.hidden = value;
    self.upArrowButton1.hidden = value;
    self.downArrayButton1.hidden = value;
    self.upArrowButton2.hidden = value;
    self.downArrayButton2.hidden = value;
    self.eventLable.hidden = value;
    self.eventLable1.hidden = value;
    self.eventLable2.hidden = value;
    self.timeLabel.hidden = value;
    self.timeLabel1.hidden = value;
    self.timeLabel2.hidden = value;
    self.navButton.hidden = value;
    self.editButton.hidden = value;
    self.submitButton.hidden = value;

}

-(void)ShowStudyingIn:(BOOL)value
{
//  CGRect frame = CGRectMake(10, 10, 130, 130);
    self.studyingView.hidden = value;
    self.tvView.hidden = value;
    self.tvView.frame =CGRectMake(20, 20, 130, 130);
    self.moviesView.hidden = value;
    self.moviesView.frame = CGRectMake(170, 20, 130, 130);;
    self.vgView.hidden = value;
    self.vgView.frame = CGRectMake(20, 170, 130, 130);
    self.otherView.hidden = value;
    self.otherView.frame = CGRectMake(170, 170, 130, 130);
    self.popularTonightButton.hidden = value;
    self.agendaButton.hidden = value;

}



-(IBAction)populatTonight:(BOOL)value
{
    PopularViewController *PopularViewControllerObj =
[[PopularViewController alloc]
initWithNibName:@"PopularViewController" bundle:nil];
    [[self
navigationController]pushViewController:PopularViewControll
erObj animated:YES];
    [PopularViewControllerObj release];
}
-(IBAction)agendaButtonMethod:(BOOL)value
{
    AgendaStayingInViewController
```

```
*AgendaStayingInViewControllerObj =
[[AgendaStayingInViewController alloc]
initWithNibName:@"AgendaStayingInViewController"
bundle:nil];
    [[self navigationController]
pushViewController:AgendaStayingInViewControllerObj
animated:YES];
    [AgendaStayingInViewControllerObj release];
}

-(IBAction)subCourseDetails:(id)sender
{

}


-(IBAction)movingButton:(id)sender
{

}





/*
// Implement viewDidLoad to do additional setup after
loading the view, typically from a nib.
- (void)viewDidLoad {
    [super viewDidLoad];
}
*/

/*
// Override to allow orientations other than the default
portrait orientation.
-
(BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOr
ientation)interfaceOrientation {
    // Return YES for supported orientations.
    return (interfaceOrientation ==
UIInterfaceOrientationPortrait);
}
*/

- (void)didReceiveMemoryWarning {
    // Releases the view if it doesn't have a superview.
    [super didReceiveMemoryWarning];

    // Release any cached data, images, etc. that aren't in
use.
```

```
}

- (void)viewDidUnload {
    [super viewDidUnload];
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}


- (void)dealloc {
    [super dealloc];
}


@end
```

```
//
//  ClassMateClass.h
//  GrowndSwell
//
//  Created by ArtiVilla on 3/29/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import <UIKit/UIKit.h>


@interface ClassMateClass : UIViewController {

    IBOutlet UISearchBar *searchBar;
    IBOutlet UITableView *classMateTable;
    IBOutlet UIImageView *topBar;

}

@property (nonatomic,retain) IBOutlet UISearchBar
*searchBar;
@property (nonatomic,retain) IBOutlet UITableView
*classMateTable;
@property (nonatomic,retain) IBOutlet UIImageView *topBar;


@end
```

```
//
//  ClassMateClass.m
//  GrowndSwell
//
//  Created by ArtiVilla on 3/29/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import "ClassMateClass.h"


@implementation ClassMateClass


@synthesize searchBar,classMateTable,topBar;

// The designated initializer.  Override if you create the
controller programmatically and want to perform
customization that is not appropriate for viewDidLoad.
/*
- (id)initWithNibName:(NSString *)nibNameOrNil
bundle:(NSBundle *)nibBundleOrNil {
    self = [super initWithNibName:nibNameOrNil
bundle:nibBundleOrNil];
    if (self) {
        // Custom initialization.
    }
    return self;
}
*/

/*
// Implement viewDidLoad to do additional setup after
loading the view, typically from a nib.
- (void)viewDidLoad {
    [super viewDidLoad];
}
*/

/*
// Override to allow orientations other than the default
portrait orientation.
-
(BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOr
ientation)interfaceOrientation {
    // Return YES for supported orientations.
    return (interfaceOrientation ==
UIInterfaceOrientationPortrait);
}
*/
```

```
- (void)didReceiveMemoryWarning {
    // Releases the view if it doesn't have a superview.
    [super didReceiveMemoryWarning];

    // Release any cached data, images, etc. that aren't in
use.
}

- (void)viewDidUnload {
    [super viewDidUnload];
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}


- (void)dealloc {
    [super dealloc];
}


@end
```

```
//
//  EditStudyingController.h
//  GrowndSwell
//
//  Created by ArtiVilla on 3/29/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import <UIKit/UIKit.h>


@interface EditStudyingController : UIViewController {

}

@end
```

```
//
//  EditStudyingController.m
//  GrowndSwell
//
//  Created by ArtiVilla on 3/29/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import "EditStudyingController.h"


@implementation EditStudyingController

// The designated initializer.  Override if you create the
controller programmatically and want to perform
customization that is not appropriate for viewDidLoad.
/*
- (id)initWithNibName:(NSString *)nibNameOrNil
bundle:(NSBundle *)nibBundleOrNil {
    self = [super initWithNibName:nibNameOrNil
bundle:nibBundleOrNil];
    if (self) {
        // Custom initialization.
    }
    return self;
}
*/

/*
// Implement viewDidLoad to do additional setup after
loading the view, typically from a nib.
- (void)viewDidLoad {
    [super viewDidLoad];
}
*/

/*
// Override to allow orientations other than the default
portrait orientation.
-
(BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOr
ientation)interfaceOrientation {
    // Return YES for supported orientations.
    return (interfaceOrientation ==
UIInterfaceOrientationPortrait);
}
*/

- (void)didReceiveMemoryWarning {
    // Releases the view if it doesn't have a superview.
    [super didReceiveMemoryWarning];

    // Release any cached data, images, etc. that aren't in 
use.
}

- (void)viewDidUnload {
    [super viewDidUnload];
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}


- (void)dealloc {
    [super dealloc];
}


@end
```

```
//
//  FriendsViewController.h
//  GrowndSwell
//
//  Created by ArtiVilla on 3/18/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import <UIKit/UIKit.h>


@interface FriendsViewController : UIViewController {

    IBOutlet UIImageView *topBarView;
    IBOutlet UISearchBar *searchBar;
    IBOutlet UIButton *everyOneButton;
    IBOutlet UIButton *groupsButton;
    IBOutlet UIButton *simAgendaButton;
    IBOutlet UIButton *allFriendsButton;
    IBOutlet UITableView * friendsTable;
    NSMutableArray *contentsArray;
    IBOutlet UIView *addFriendsView;

    IBOutlet UIButton *twitterButton;
    IBOutlet UIButton *emailButton;
    IBOutlet UIButton *faceBookButton;
    IBOutlet UIButton *creatGroupButton;

    NSMutableArray *friendsArray;
    NSMutableArray *groupsArray;
    NSMutableArray *saFriendsArray;
    NSMutableArray *dataToShow;


}

@property (nonatomic,retain) IBOutlet UIButton
*creatGroupButton;
@property (nonatomic,retain) IBOutlet UIButton
*twitterButton;
@property (nonatomic,retain) IBOutlet UIButton
*emailButton;
@property (nonatomic,retain) IBOutlet UIButton
*faceBookButton;
@property (nonatomic,retain) IBOutlet UIImageView
*topBarView;
@property (nonatomic,retain) IBOutlet UIView
*addFriendsView;
@property (nonatomic,retain) IBOutlet UISearchBar
*searchBar;
```

```
@property (nonatomic,retain) IBOutlet UIButton
*everyOneButton;
@property (nonatomic,retain) IBOutlet UIButton
*groupsButton;
@property (nonatomic,retain) IBOutlet UIButton
*simAgendaButton;
@property (nonatomic,retain) IBOutlet UIButton
*allFriendsButton;
@property (nonatomic,retain) IBOutlet UITableView *
friendsTable;

-(IBAction)addFriends;
-(IBAction)showEveryOne;
-(IBAction)showGroups;
-(IBAction)similarAgenda;
-(IBAction)showFaceBook;
-(IBAction)showTwitter;;
-(IBAction)showContacts;
-(IBAction)showAllView;
-(IBAction)createGroups;
-(id) initWithTabBar;




@end
```

```
//
//  FriendsViewController.m
//  GrowndSwell
//
//  Created by ArtiVilla on 3/18/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import "FriendsViewController.h"


@implementation FriendsViewController
@synthesize
topBarView,searchBar,everyOneButton,groupsButton,simAgendaB
utton,allFriendsButton,friendsTable;
@synthesize addFriendsView;
@synthesize
twitterButton,creatGroupButton,emailButton,faceBookButton;

-(id) initWithTabBar
{
    self = [super init];
    if (self)
    {
     //this is the label on the tab button itself
     self.title = @"Friends";
     //parsingType = [[NSString
alloc]initWithString:@"recent"];
     //Image of the Tab
     //loadStatus = @"loading";
     self.tabBarItem.image = [UIImage
imageNamed:@"Places.png"];
    }
    return self;
}


-(void)viewWillAppear:(BOOL)animated
{
    self.navigationController.navigationBarHidden = YES;
}


// The designated initializer.  Override if you create the
controller programmatically and want to perform
customization that is not appropriate for viewDidLoad.
/*
- (id)initWithNibName:(NSString *)nibNameOrNil
bundle:(NSBundle *)nibBundleOrNil {
    self = [super initWithNibName:nibNameOrNil
bundle:nibBundleOrNil];
    if (self) {
        // Custom initialization.
    }
    return self;
}
*/


// Implement viewDidLoad to do additional setup after
loading the view, typically from a nib.
- (void)viewDidLoad {
    [super viewDidLoad];
    friendsArray = [[NSMutableArray alloc] init];
    groupsArray = [[NSMutableArray alloc] init];
    saFriendsArray = [[NSMutableArray alloc] init];
    dataToShow = [[NSMutableArray alloc] init];
    [self showEveryOne];
}


/*
// Override to allow orientations other than the default
portrait orientation.
-
(BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOr
ientation)interfaceOrientation {
    // Return YES for supported orientations.
    return (interfaceOrientation ==
UIInterfaceOrientationPortrait);
}
*/



- (NSInteger)numberOfSectionsInTableView:(UITableView
*)tableView              // Default is 1 if not implemented
{
    return 1;
}

- (NSInteger)tableView:(UITableView *)tableView
numberOfRowsInSection:(NSInteger)section
{
    return [dataToShow count];
}

// Row display. Implementers should *always* try to reuse
cells by setting each cell's reuseIdentifier and querying
for available reusable cells with
```

```
dequeueReusableCellWithIdentifier:
// Cell gets various attributes set automatically based on
table (separators) and data source (accessory views,
editing controls)

- (UITableViewCell *)tableView:(UITableView *)tableView
cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    //static NSString *CellIdentifier = @"Cell";

    UITableViewCell *cell = [friendsTable
dequeueReusableCellWithIdentifier:nil];
    if (cell == nil) {
        cell = [[UITableViewCell alloc]
initWithStyle:UITableViewCellStyleDefault
reuseIdentifier:nil];
    }

    // Configure the cell...
   // cell.textLabel.text = @"Test";
    [cell addSubview:[dataToShow
objectAtIndex:indexPath.row]];
    //cell.backgroundColor = [UIColor whiteColor];
    cell.selectionStyle =
UITableViewCellSelectionStyleNone;
    return cell;
}


- (CGFloat)tableView:(UITableView *)tableView
heightForRowAtIndexPath:(NSIndexPath *)indexPath
{
    return ((UIView*)[dataToShow
objectAtIndex:indexPath.row]).frame.size.height;
}


-(IBAction)showAllView
{
    addFriendsView.hidden = NO;

}
-(IBAction)showGroups
{
    addFriendsView.hidden = YES;


    if(dataToShow!=nil)
      [dataToShow release];
    for(int counter = 0; counter<10; counter++)
    {
     UIView *tempView = [[UIView alloc]
initWithFrame:CGRectMake(0, 0, 320, 135)];
     tempView.backgroundColor = [UIColor blueColor];

     UILabel *groupName = [[UILabel alloc]
initWithFrame:CGRectMake(10, 10, 100, 25)];
     groupName.text = @"Group Name";
     groupName.backgroundColor = [UIColor clearColor];
     [tempView addSubview:groupName];
     [groupName release];

     UIImageView *groupImage = [[UIImageView alloc]
initWithFrame:CGRectMake(10, 30, 150, 100)];
     [groupName setBackgroundColor:[UIColor redColor]];
     groupImage.image = [UIImage
imageNamed:@"backGround.png"];
     [tempView addSubview:groupImage];
     [groupImage release];

     UILabel *numberOfUsers = [[UILabel alloc]
initWithFrame:CGRectMake(210, 65, 100, 25)];
     numberOfUsers.text = [NSString stringWithFormat: @"%d
Peoples",counter];
     numberOfUsers.backgroundColor = [UIColor clearColor];
     [tempView addSubview:numberOfUsers];
     [numberOfUsers release];

     [groupsArray addObject:tempView];
     [tempView release];

    }

    dataToShow = [[NSMutableArray alloc]
initWithArray:groupsArray];
    self.friendsTable.frame = CGRectMake(0, 85, 320, 280);
    self.creatGroupButton.hidden = NO;
    [self.friendsTable reloadData];
}


-(IBAction)showEveryOne
{
    addFriendsView.hidden = YES;
    if(dataToShow!=nil)
     [dataToShow release];

    for(int counter = 0; counter<20; counter++)
    {
      UIView *tempView = [[UIView alloc]
initWithFrame:CGRectMake(0, 0, 320, 35)];
      tempView.backgroundColor = [UIColor blueColor];

      UILabel *groupName = [[UILabel alloc]
initWithFrame:CGRectMake(10, 2, 320, 30)];
      groupName.text = [NSString stringWithFormat: @"Friends
Name %d",counter];
      groupName.backgroundColor = [UIColor clearColor];
      [tempView addSubview:groupName];
      [groupName release];

      /*UIImageView *groupImage = [[UIImageView alloc]
initWithFrame:CGRectMake(10, 30, 150, 100)];
      [groupName setBackgroundColor:[UIColor redColor]];
      groupImage.image = [UIImage
imageNamed:@"backGround.png"];
      [tempView addSubview:groupName];
      [groupImage release];

      UILabel *numberOfUsers = [[UILabel alloc]
initWithFrame:CGRectMake(210, 65, 100, 25)];
      numberOfUsers.text = [NSString stringWithFormat: @"%d
Peoples",counter];
      numberOfUsers.backgroundColor = [UIColor clearColor];
      [tempView addSubview:numberOfUsers];
      [numberOfUsers release];
      **/
      [friendsArray addObject:tempView];
      [tempView release];

    }

    dataToShow = [[NSMutableArray alloc]
initWithArray:friendsArray];
    self.friendsTable.frame = CGRectMake(0, 85, 320, 335);
    self.creatGroupButton.hidden = YES;
    [self.friendsTable reloadData];
}

-(IBAction)createGroups
{

}

-(IBAction)similarAgenda
{
    addFriendsView.hidden = YES;
    self.creatGroupButton.hidden = NO;
    if(dataToShow!=nil)
     [dataToShow release];

    for(int counter = 0; counter<10; counter++)
    {
     UIView *tempView = [[UIView alloc]
initWithFrame:CGRectMake(0, 0, 320, 135)];
     tempView.backgroundColor = [UIColor blueColor];

     UILabel *groupName = [[UILabel alloc]
initWithFrame:CGRectMake(10, 10, 100, 25)];
     groupName.text = [NSString stringWithFormat:@"Class
Mate %d",counter];
     groupName.backgroundColor = [UIColor clearColor];
     [tempView addSubview:groupName];
     [groupName release];

     UIImageView *groupImage = [[UIImageView alloc]
initWithFrame:CGRectMake(10, 30, 150, 100)];
     [groupName setBackgroundColor:[UIColor redColor]];
     groupImage.image = [UIImage
imageNamed:@"backGround.png"];
     [tempView addSubview:groupImage];
     [groupImage release];

     UILabel *firstEvent = [[UILabel alloc]
initWithFrame:CGRectMake(210, 15, 100, 25)];
     firstEvent.text = [NSString stringWithFormat: @"%d
Peoples",counter];
     firstEvent.backgroundColor = [UIColor clearColor];
     [tempView addSubview:firstEvent];
     [firstEvent release];


     UILabel *secondEvent = [[UILabel alloc]
initWithFrame:CGRectMake(210, 45, 100, 25)];
     secondEvent.text = [NSString stringWithFormat: @"%d
Peoples",counter];
     secondEvent.backgroundColor = [UIColor clearColor];
     [tempView addSubview:secondEvent];
     [secondEvent release];


     UILabel *thirdEvent = [[UILabel alloc]
initWithFrame:CGRectMake(210, 75, 100, 25)];
     thirdEvent.text = [NSString stringWithFormat: @"%d
Peoples",counter];
     thirdEvent.backgroundColor = [UIColor clearColor];
     [tempView addSubview:thirdEvent];
     [thirdEvent release];

     [saFriendsArray addObject:tempView];
     [tempView release];

    }

    dataToShow = [[NSMutableArray alloc]
initWithArray:saFriendsArray];
    self.friendsTable.frame = CGRectMake(0, 85, 320, 335);
    self.creatGroupButton.hidden = YES;
    [self.friendsTable reloadData];

}


-(IBAction)showFaceBook
{
}
-(IBAction)showTwitter
{
}
-(IBAction)showContacts
{
}




- (void)didReceiveMemoryWarning {
    // Releases the view if it doesn't have a superview.
    [super didReceiveMemoryWarning];

    // Release any cached data, images, etc. that aren't in
use.
}

- (void)viewDidUnload {
    [super viewDidUnload];
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}


- (void)dealloc {
    [super dealloc];
}


@end
```

```
//
//  GrowndSwellAppDelegate.h
//  GrowndSwell
//
//  Created by ArtiVilla on 3/9/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import <UIKit/UIKit.h>
#import "PlacesViewController.h"
#import "AgendaViewController.h"
#import "FriendsViewController.h"
#import "SwellsViewController.h"


@class GrowndSwellViewController;

@interface GrowndSwellAppDelegate : NSObject
<UIApplicationDelegate> {
    UIWindow *window;
    GrowndSwellViewController *viewController;
    UITabBarController *bottomTabBar;
    UINavigationController *navigationController;
    AgendaViewController *AgendaViewControllerObj;
}

@property (nonatomic, retain) IBOutlet UIWindow *window;
@property (nonatomic, retain) UITabBarController
*bottomTabBar;
@property (nonatomic, retain) IBOutlet
GrowndSwellViewController *viewController;
@property (nonatomic, retain) UINavigationController
*navigationController;

-(void) loadTabBar;


@end
```

```
//
//  GrowndSwellAppDelegate.m
//  GrowndSwell
//
//  Created by ArtiVilla on 3/9/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import "GrowndSwellAppDelegate.h"
#import "GrowndSwellViewController.h"

@implementation GrowndSwellAppDelegate

@synthesize window;
@synthesize
bottomTabBar,viewController,navigationController;


#pragma mark -
#pragma mark Application lifecycle

- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{

    // Override point for customization after application
launch.

    // Add the view controller's view to the window and
display.
    navigationController = [[UINavigationController
alloc]initWithRootViewController:viewController];

    [self.window addSubview:navigationController.view];
    [self.window makeKeyAndVisible];

    return YES;
}

-(void)loadSubView:(UIViewController*)viewController
{
    [[self navigationController]
pushViewController:viewController animated:YES];
}

- (void)applicationWillResignActive:(UIApplication
*)application {
    /*
     Sent when the application is about to move from active
to inactive state. This can occur for certain types of
```

```
temporary interruptions (such as an incoming phone call or
SMS message) or when the user quits the application and it
begins the transition to the background state.
     Use this method to pause ongoing tasks, disable
timers, and throttle down OpenGL ES frame rates. Games
should use this method to pause the game.
     */
}


//###########################################################
##############################
//Method for loading tab bar after login
//###########################################################
##############################


-(void) loadTabBar
{
    //load tab bar
    bottomTabBar = [[UITabBarController alloc]init];
    NSMutableArray *controllersArray = [[NSMutableArray
alloc]initWithCapacity:5];
    // Creating TabBarController with Navigation Controller
    UINavigationController *navigatController;

    AgendaViewControllerObj = [[AgendaViewController
alloc]initWithTabBar];
    navigatController = [[UINavigationController
alloc]initWithRootViewController:AgendaViewControllerObj];
    [navigatController.navigationBar
setBarStyle:UIBarStyleBlackOpaque];
    [controllersArray addObject:navigatController];
    [AgendaViewControllerObj release];
    [navigatController release];

    SwellsViewController *SwellsViewControllerObj =
[[SwellsViewController alloc]initWithTabBar];
    navigatController = [[UINavigationController
alloc]initWithRootViewController:SwellsViewControllerObj];
    [navigatController.navigationBar
setBarStyle:UIBarStyleBlackOpaque];
    [controllersArray addObject:navigatController];
    [SwellsViewControllerObj release];
    [navigatController release];


    PlacesViewController *PlacesViewControllerObj =
[[PlacesViewController alloc]initWithTabBar];
    navigatController = [[UINavigationController
alloc]initWithRootViewController:PlacesViewControllerObj];
    [navigatController.navigationBar 
setBarStyle:UIBarStyleBlackOpaque];
    [controllersArray addObject:navigatController];
    [PlacesViewControllerObj release];
    [navigatController release];

    FriendsViewController *friendsView = 
[[FriendsViewController alloc]initWithTabBar];
    navigatController = [[UINavigationController 
alloc]initWithRootViewController:friendsView];
    [navigatController.navigationBar 
setBarStyle:UIBarStyleBlackOpaque];
    [controllersArray addObject:navigatController];
    [friendsView release];
    [navigatController release];


/*  MoreViewController *moreViewController = 
[[MoreViewController alloc]initWithTabBar];
    navigatController = [[UINavigationController 
alloc]initWithRootViewController:moreViewController];
    [navigatController.navigationBar 
setBarStyle:UIBarStyleBlackOpaque];
    [controllersArray addObject:navigatController];
    [moreViewController release];
    [navigatController release];
    */

    //Asigning controllers array to TabBarController
    bottomTabBar.viewControllers = controllersArray;
    [controllersArray release];
    window.backgroundColor = [UIColor 
groupTableViewBackgroundColor];

    // Add the tab bar controller's current view as a 
subview of the window
    //tabBarController.view.alpha = 0.0;
    [window addSubview:bottomTabBar.view];


    //tabBarController.view.alpha = 1.0;
    [navigationController.view removeFromSuperview];

}


-(void) removeTabBar
{
    [bottomTabBar.view removeFromSuperview];
    [window addSubview:navigationController.view];


    //tabBarController.view.alpha = 1.0;
    //[navigationController.view removeFromSuperview];

}



- (void)applicationDidEnterBackground:(UIApplication 
*)application {
    /*
     Use this method to release shared resources, save user 
data, invalidate timers, and store enough application state 
information to restore your application to its current 
state in case it is terminated later. 
     If your application supports background execution, 
called instead of applicationWillTerminate: when the user 
quits.
     */
}


- (void)applicationWillEnterForeground:(UIApplication 
*)application {
    /*
     Called as part of  transition from the background to 
the inactive state: here you can undo many of the changes 
made on entering the background.
     */
}


- (void)applicationDidBecomeActive:(UIApplication 
*)application {
    /*
     Restart any tasks that were paused (or not yet 
started) while the application was inactive. If the 
application was previously in the background, optionally 
refresh the user interface.
     */
}


- (void)applicationWillTerminate:(UIApplication 
*)application {
    /*
     Called when the application is about to terminate.
     See also applicationDidEnterBackground:.
     */
}


#pragma mark -
#pragma mark Memory management

- (void)applicationDidReceiveMemoryWarning:(UIApplication
*)application {
    /*
     Free up as much memory as possible by purging cached
data objects that can be recreated (or reloaded from disk)
later.
     */
}


- (void)dealloc {
    [viewController release];
    [window release];
    [super dealloc];
}


@end
```

```
//
//  GrowndSwellViewController.h
//  GrowndSwell
//
//  Created by ArtiVilla on 3/9/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import <UIKit/UIKit.h>
#import "RegisterViewController.h"

@interface GrowndSwellViewController : UIViewController
<UITextFieldDelegate> {

    IBOutlet UITextField *userNameField;
    IBOutlet    UITextField *passwordField;
    IBOutlet UIImageView *backGroundImage;
    IBOutlet UIButton *loginButton;
    IBOutlet UIButton *signupButton;
}

//@property (nonatomic,retain) UITabBarController
*bottomTabBar;
@property (nonatomic,retain) IBOutlet UITextField
*userNameField;
@property (nonatomic,retain) IBOutlet UITextField
*passwordField;
@property (nonatomic,retain) IBOutlet UIButton
*loginButton;
@property (nonatomic,retain) IBOutlet UIButton
*signupButton;
@property (nonatomic,retain) IBOutlet UIImageView
*backGroundImage;


//Instance method
-(IBAction)signUpMethod:(id)sender;
-(IBAction)loginMethod:(id)sender;



@end
```

```
//
//  GrowndSwellViewController.m
//  GrowndSwell
//
//  Created by ArtiVilla on 3/9/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import "GrowndSwellViewController.h"
#import "GrowndSwellAppDelegate.h"

@implementation GrowndSwellViewController

@synthesize
passwordField,userNameField,backGroundImage,loginButton,sig
nupButton;

/*
// The designated initializer. Override to perform setup
that is required before the view is loaded.
- (id)initWithNibName:(NSString *)nibNameOrNil
bundle:(NSBundle *)nibBundleOrNil {
    self = [super initWithNibName:nibNameOrNil
bundle:nibBundleOrNil];
    if (self) {
        // Custom initialization
    }
    return self;
}
*/

/*
// Implement loadView to create a view hierarchy
programmatically, without using a nib.
- (void)loadView {
}
*/



// Implement viewDidLoad to do additional setup after
loading the view, typically from a nib.
- (void)viewDidLoad {
    [super viewDidLoad];
    userNameField.delegate = self;
    passwordField.delegate = self;
}

-(void)viewWillAppear:(BOOL)animated
{
    self.navigationController.navigationBarHidden = YES;
}
-(IBAction)loginMethod:(id)sender
{
    //Login logic goes here

    NSLog(@"login up");

    GrowndSwellAppDelegate *GrowndSwellAppDelegateObj =
(GrowndSwellAppDelegate*)[[UIApplication sharedApplication]
delegate];
    [GrowndSwellAppDelegateObj loadTabBar];
}



- (BOOL)textFieldShouldReturn:(UITextField *)textField
// called when 'return' key pressed. return NO to ignore.
{
    if(textField == userNameField)
     [passwordField becomeFirstResponder];
    else
     [passwordField resignFirstResponder];
    return YES;
}

- (void)textFieldDidBeginEditing:(UITextField *)textField
// became first responder
{

}
-(IBAction)signUpMethod:(id)sender
{
    //Login logic goes here
    NSLog(@"sign up");
    RegisterViewController *RegisterViewControllerObj =
[[RegisterViewController alloc]
initWithNibName:@"RegisterViewController" bundle:nil];
    [[self navigationController]
pushViewController:RegisterViewControllerObj animated:YES];

//  GrowndSwellAppDelegate *GrowndSwellAppDelegateObj =
(GrowndSwellAppDelegate*)[[UIApplication sharedApplication]
delegate];
//  [GrowndSwellAppDelegateObj
loadSubView:RegisterViewControllerObj];
    [RegisterViewControllerObj release];
}
/*
// Override to allow orientations other than the default
```

```
portrait orientation.
-
(BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOr
ientation)interfaceOrientation {
    // Return YES for supported orientations
    return (interfaceOrientation ==
UIInterfaceOrientationPortrait);
}
*/

- (void)didReceiveMemoryWarning {
    // Releases the view if it doesn't have a superview.
    [super didReceiveMemoryWarning];

    // Release any cached data, images, etc that aren't in
use.
}

- (void)viewDidUnload {
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}


- (void)dealloc {
    [super dealloc];
}

@end
```

```
//
//  MapViewController.h
//  GrowndSwell
//
//  Created by ArtiVilla on 4/3/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import <UIKit/UIKit.h>
#import <MapKit/MapKit.h>
#import <MapKit/MKReverseGeocoder.h>
#import <CoreLocation/CoreLocation.h>


@interface MapViewController : UIViewController<
MKMapViewDelegate, MKReverseGeocoderDelegate,
CLLocationManagerDelegate> {

    IBOutlet  MKMapView *mapView;
    IBOutlet  UIButton *doneButton;
    MKReverseGeocoder *geoCoder;
    MKPlacemark *mPlacemark;
    CLLocationCoordinate2D location;


}
@property (nonatomic,retain) IBOutlet  MKMapView *mapView;
@property (nonatomic,retain) IBOutlet  UIButton
*doneButton;


-(IBAction)goBack;
@end
```

```
//
//  MapViewController.m
//  GrowndSwell
//
//  Created by ArtiVilla on 4/3/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import "MapViewController.h"


@implementation MapViewController
@synthesize mapView,doneButton;
// The designated initializer.  Override if you create the
controller programmatically and want to perform
customization that is not appropriate for viewDidLoad.
/*
- (id)initWithNibName:(NSString *)nibNameOrNil
bundle:(NSBundle *)nibBundleOrNil {
    self = [super initWithNibName:nibNameOrNil
bundle:nibBundleOrNil];
    if (self) {
        // Custom initialization.
    }
    return self;
}
*/


// Implement viewDidLoad to do additional setup after
loading the view, typically from a nib.
- (void)viewDidLoad {

     self.mapView.showsUserLocation = YES;
     self.mapView.delegate = self;
     self.mapView.hidden = NO;

     //    Region and Zoom
     MKCoordinateRegion region;
     MKCoordinateSpan span;
     span.latitudeDelta=0.2;
     span.longitudeDelta=0.2;

     CLLocationManager *locationManager=[[CLLocationManager
alloc] init];
     locationManager.delegate=self;

locationManager.desiredAccuracy=kCLLocationAccuracyNearestT
enMeters;
```

```
     [locationManager startUpdatingLocation];


     /*location=mapView.userLocation.coordinate;
     location.latitude=40.814849;
     location.longitude=-73.622732;
     region.span=span;
     region.center=location;*/

     /*Geocoder Stuff*/

     geoCoder=[[MKReverseGeocoder alloc]
initWithCoordinate:location];
     geoCoder.delegate=self;
     [geoCoder start];


     [mapView setRegion:region animated:TRUE];
     [mapView regionThatFits:region];
     [super viewDidLoad];
}



- (void)reverseGeocoder:(MKReverseGeocoder *)geocoder
didFailWithError:(NSError *)error{
    NSLog(@"Reverse Geocoder Errored");

}

- (void)reverseGeocoder:(MKReverseGeocoder *)geocoder
didFindPlacemark:(MKPlacemark *)placemark{
    NSLog(@"Reverse Geocoder completed");
//  mPlacemark=placemark;
    [mapView addAnnotation:placemark];
}

- (MKAnnotationView *) mapView:(MKMapView *)mapView
viewForAnnotation:(id <MKAnnotation>) annotation{
    MKPinAnnotationView *annView=[[MKPinAnnotationView
alloc] initWithAnnotation:annotation
reuseIdentifier:@"currentloc"];
    annView.animatesDrop=TRUE;
    return annView;
}

- (void)locationManager:(CLLocationManager *)manager
didUpdateToLocation:(CLLocation *)newLocation
fromLocation:(CLLocation *)oldLocation{
    location=newLocation.coordinate;
    //One location is obtained.. just zoom to that location
    NSLog(@"Reverse locationManager completed");

    MKCoordinateRegion region;
    region.center=location;
    //Set Zoom level using Span
    MKCoordinateSpan span;
    span.latitudeDelta=.005;
    span.longitudeDelta=.005;
    region.span=span;

    [self.mapView setRegion:region animated:TRUE];

}

- (void)locationManager:(CLLocationManager *)manager
didFailWithError:(NSError *)error{
}





/*
// Override to allow orientations other than the default
portrait orientation.
-
(BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOr
ientation)interfaceOrientation {
    // Return YES for supported orientations.
    return (interfaceOrientation ==
UIInterfaceOrientationPortrait);
}
*/
-(IBAction)goBack
{
    [[self navigationController]
popViewControllerAnimated:YES];
}

- (void)didReceiveMemoryWarning {
    // Releases the view if it doesn't have a superview.
    [super didReceiveMemoryWarning];

    // Release any cached data, images, etc. that aren't in
use.
}

- (void)viewDidUnload {
    [super viewDidUnload];
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}


- (void)dealloc {
    [super dealloc];
}


@end
```

```
//
//  PeopleHereViewController.h
//  GrowndSwell
//
//  Created by ArtiVilla on 3/28/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import <UIKit/UIKit.h>
#import "ProfilePageViewController.h"

@interface PeopleHereViewController : UIViewController {

    IBOutlet UISearchBar *searchBar;
    IBOutlet UITableView *tableView;
    NSMutableArray *dataToShow;
    ProfilePageViewController
*ProfilePageViewControllerObj;
}


@property (nonatomic,retain) IBOutlet UISearchBar
*searchBar;
@property (nonatomic,retain) IBOutlet UITableView
*tableView;

@end
```

```
//
//  PeopleHereViewController.m
//  GrowndSwell
//
//  Created by ArtiVilla on 3/28/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import "PeopleHereViewController.h"



@implementation PeopleHereViewController
@synthesize searchBar,tableView;

// The designated initializer.  Override if you create the
controller programmatically and want to perform
customization that is not appropriate for viewDidLoad.
/*
- (id)initWithNibName:(NSString *)nibNameOrNil
bundle:(NSBundle *)nibBundleOrNil {
    self = [super initWithNibName:nibNameOrNil
bundle:nibBundleOrNil];
    if (self) {
        // Custom initialization.
    }
    return self;
}
*/


// Implement viewDidLoad to do additional setup after
loading the view, typically from a nib.
- (void)viewDidLoad {
    [super viewDidLoad];

    dataToShow = [[NSMutableArray alloc] init];
    [self showFriends];

}


/*
// Override to allow orientations other than the default
portrait orientation.
-
(BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOr
ientation)interfaceOrientation {
    // Return YES for supported orientations.
    return (interfaceOrientation ==
UIInterfaceOrientationPortrait);
}
*/




-(IBAction)showFriends
{

    if([dataToShow count]>0)
     [dataToShow removeAllObjects];

    for(int counter = 0; counter<10; counter++)
    {
     UIView *tempView = [[UIView alloc]
initWithFrame:CGRectMake(0, 0, 320, 110)];
     tempView.backgroundColor = [UIColor blueColor];

     UIImageView *groupImage = [[UIImageView alloc]
initWithFrame:CGRectMake(10, 5, 120, 100)];
     groupImage.image = [UIImage
imageNamed:@"backGround.png"];
     [tempView addSubview:groupImage];
     [groupImage release];

     UILabel *numberOfUsers = [[UILabel alloc]
initWithFrame:CGRectMake(160, 37, 100, 25)];
     numberOfUsers.text = [NSString stringWithFormat:
@"Friend Name  %d",counter];
     numberOfUsers.backgroundColor = [UIColor clearColor];
     [tempView addSubview:numberOfUsers];
     [numberOfUsers release];

     UIButton *addFriend = [UIButton
buttonWithType:UIButtonTypeRoundedRect];
     [addFriend addTarget:self action:@selector(addFriend)
forControlEvents:UIControlEventTouchUpInside];
     [tempView addSubview:addFriend];
     addFriend.frame = CGRectMake(275, 35, 40, 30);
     tempView.tag = counter;
     [addFriend setTitle:@"Add"
forState:UIControlStateNormal];

     [dataToShow addObject:tempView];
     [tempView release];

    }

    [self.tableView reloadData];
```

```
}



- (NSInteger)numberOfSectionsInTableView:(UITableView
*)tableView       // Default is 1 if not implemented
{
    return 1;
}

- (NSInteger)tableView:(UITableView *)tableView
numberOfRowsInSection:(NSInteger)section
{
    return [dataToShow count];
}

// Row display. Implementers should *always* try to reuse
cells by setting each cell's reuseIdentifier and querying
for available reusable cells with
dequeueReusableCellWithIdentifier:
// Cell gets various attributes set automatically based on
table (separators) and data source (accessory views,
editing controls)

- (UITableViewCell *)tableView:(UITableView *)tableView
cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    //static NSString *CellIdentifier = @"Cell";

    UITableViewCell *cell = [tableView
dequeueReusableCellWithIdentifier:nil];
    if (cell == nil) {
        cell = [[UITableViewCell alloc]
initWithStyle:UITableViewCellStyleDefault
reuseIdentifier:nil];
    }

    // Configure the cell...
    // cell.textLabel.text = @"Test";
    [cell addSubview:[dataToShow
objectAtIndex:indexPath.row]];
    //cell.backgroundColor = [UIColor whiteColor];
    cell.selectionStyle =
UITableViewCellSelectionStyleNone;
    return cell;
}


- (CGFloat)tableView:(UITableView *)tableView
```

```
heightForRowAtIndexPath:(NSIndexPath *)indexPath
{
    return ((UIView*)[dataToShow
objectAtIndex:indexPath.row]).frame.size.height;
}




- (void)tableView:(UITableView *)tableView
didSelectRowAtIndexPath:(NSIndexPath *)indexPath
{
    ProfilePageViewControllerObj =
[[ProfilePageViewController alloc] init];
    [[self navigationController]
pushViewController:ProfilePageViewControllerObj
animated:YES];
    [ProfilePageViewControllerObj release];
}
- (void)didReceiveMemoryWarning {
    // Releases the view if it doesn't have a superview.
    [super didReceiveMemoryWarning];

    // Release any cached data, images, etc. that aren't in
use.
}

- (void)viewDidUnload {
    [super viewDidUnload];
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}


- (void)dealloc {
    [super dealloc];
}


@end
```

```
//
//  PlacesViewController.h
//  GrowndSwell
//
//  Created by ArtiVilla on 3/18/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import <UIKit/UIKit.h>


@interface PlacesViewController : UIViewController {

    IBOutlet UIView *addVenueBack;
    IBOutlet UITextField *nameField;
    IBOutlet UITextField *addField;
    IBOutlet UITextField *csField;
    IBOutlet UITextField *cityField;
    IBOutlet UITextField *stateField;
    IBOutlet UITextField *zipField;
    IBOutlet UITextField *phoneField;
    IBOutlet UITableView *venueTable;
    IBOutlet UISearchBar *venueSearch;

    IBOutlet UIButton *treddingButton;
    IBOutlet UIButton *favButton;
    IBOutlet UIButton *nearByButton;
    IBOutlet UIButton *addVenueButton;
    IBOutlet UIButton *subAddVenueButton;

    NSMutableArray *treddingArray;
    NSMutableArray *favArray;
    NSMutableArray *nearByArray;
    NSMutableArray *dataToShowArray;


}

@property (nonatomic,retain) IBOutlet UIButton
*subAddVenueButton;
@property (nonatomic,retain) IBOutlet UIButton
*treddingButton;
@property (nonatomic,retain) IBOutlet UIButton *favButton;
@property (nonatomic,retain) IBOutlet UIButton
*nearByButton;
@property (nonatomic,retain) IBOutlet UIButton
*addVenueButton;

@property (nonatomic,retain) IBOutlet UISearchBar
*venueSearch;
```

```
@property (nonatomic,retain) IBOutlet UIView *addVenueBack;
@property (nonatomic,retain) IBOutlet UITableView
*venueTable;
@property (nonatomic,retain) IBOutlet UITextField
*nameField;;
@property (nonatomic,retain) IBOutlet UITextField
*addField;;
@property (nonatomic,retain) IBOutlet UITextField
*csField;;
@property (nonatomic,retain) IBOutlet UITextField
*cityField;;
@property (nonatomic,retain) IBOutlet UITextField
*stateField;;
@property (nonatomic,retain) IBOutlet UITextField
*zipField;;
@property (nonatomic,retain) IBOutlet UITextField
*phoneField;;


///methods
-(IBAction)tredding;
-(IBAction)favVenue;
-(IBAction)nearBy;
-(IBAction)addVenue;
-(IBAction)addVenueCall;


@end
```

```
//
//  PlacesViewController.m
//  GrowndSwell
//
//  Created by ArtiVilla on 3/18/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import "PlacesViewController.h"
#import "VenueDetailsController.h"

@implementation PlacesViewController

@synthesize venueSearch,venueTable,addVenueBack,addField,
nameField,csField, cityField,stateField,
zipField,phoneField;
@synthesize
subAddVenueButton,treddingButton,favButton,nearByButton,add
VenueButton;



-(id) initWithTabBar
{
    self = [super init];
    if (self)
    {
     //this is the label on the tab button itself
     self.title = @"Places";
     //parsingType = [[NSString
alloc]initWithString:@"recent"];
     //Image of the Tab
     //loadStatus = @"loading";
     self.tabBarItem.image = [UIImage
imageNamed:@"Places.png"];
    }
    return self;
}

-(void)viewWillAppear:(BOOL)animated
{
    self.navigationController.navigationBarHidden = YES;
}
// The designated initializer.  Override if you create the
controller programmatically and want to perform
customization that is not appropriate for viewDidLoad.
/*
- (id)initWithNibName:(NSString *)nibNameOrNil
bundle:(NSBundle *)nibBundleOrNil {
    self = [super initWithNibName:nibNameOrNil
```

```
bundle:nibBundleOrNil];
    if (self) {
        // Custom initialization.
    }
    return self;
}
*/


// Implement viewDidLoad to do additional setup after
loading the view, typically from a nib.
- (void)viewDidLoad {
    [super viewDidLoad];

    treddingArray = [[NSMutableArray alloc] init];
    favArray = [[NSMutableArray alloc] init];
    nearByArray = [[NSMutableArray alloc] init];
    dataToShowArray = [[NSMutableArray alloc] init];
    [self tredding];
}




-(IBAction)tredding
{
    if(dataToShowArray!=nil)
     [dataToShowArray release];

    for(int counter = 0; counter<10; counter++)
    {
     UIView *tempView = [[UIView alloc]
initWithFrame:CGRectMake(0, 0, 320, 135)];
     tempView.backgroundColor = [UIColor blueColor];

     UIImageView *groupImage = [[UIImageView alloc]
initWithFrame:CGRectMake(10, 10, 150, 100)];
     [groupImage setBackgroundColor:[UIColor redColor]];
     groupImage.image = [UIImage
imageNamed:@"backGround.png"];
     [tempView addSubview:groupImage];
     [groupImage release];

     UILabel *numberOfUsers = [[UILabel alloc]
initWithFrame:CGRectMake(175, 65, 100, 25)];
     numberOfUsers.text = [NSString stringWithFormat: @"%d
Peoples",counter];
     numberOfUsers.backgroundColor = [UIColor clearColor];
     [tempView addSubview:numberOfUsers];
     [numberOfUsers release];
     UIImageView *starImage = [[UIImageView alloc]
initWithFrame:CGRectMake(285, 50, 30, 30)];
     [starImage setBackgroundColor:[UIColor redColor]];
     starImage.image = [UIImage
imageNamed:@"backGround.png"];
     [tempView addSubview:starImage];
     [starImage release];


     [treddingArray addObject:tempView];
     [tempView release];

    }
    self.venueTable.hidden = NO;

    dataToShowArray = [[NSMutableArray alloc]
initWithArray:treddingArray];
    self.addVenueBack.hidden = YES;
    [self.venueTable reloadData];

}


-(IBAction)favVenue
{

    if(dataToShowArray!=nil)
     [dataToShowArray release];

    for(int counter = 0; counter<10; counter++)
    {
     UIView *tempView = [[UIView alloc]
initWithFrame:CGRectMake(0, 0, 320, 135)];
     tempView.backgroundColor = [UIColor blueColor];

     UIImageView *groupImage = [[UIImageView alloc]
initWithFrame:CGRectMake(10, 5, 130, 90)];
     [groupImage setBackgroundColor:[UIColor redColor]];
     groupImage.image = [UIImage
imageNamed:@"backGround.png"];
     [tempView addSubview:groupImage];
     [groupImage release];

     UILabel *venueName= [[UILabel alloc]
initWithFrame:CGRectMake(10, 105, 100, 25)];
     venueName.text = [NSString stringWithFormat: @"Venue
name"];
     venueName.backgroundColor = [UIColor clearColor];
     [tempView addSubview:venueName];
     [venueName release];

     UILabel *ranking = [[UILabel alloc]
initWithFrame:CGRectMake(175, 65, 80, 25)];
     ranking.text = [NSString stringWithFormat: @"Ranking
%d",counter];
     ranking.backgroundColor = [UIColor clearColor];
     [tempView addSubview:ranking];
     [ranking release];

     UILabel *capcity = [[UILabel alloc]
initWithFrame:CGRectMake(275, 65, 100, 25)];
     capcity.text = [NSString stringWithFormat: @"Capcity
%d ",counter];
     capcity.backgroundColor = [UIColor clearColor];
     [tempView addSubview:capcity];
     [capcity release];

     [favArray addObject:tempView];
     [tempView release];

    }
    self.venueTable.hidden = NO;

    dataToShowArray = [[NSMutableArray alloc]
initWithArray:favArray];
    self.addVenueBack.hidden = YES;
    [self.venueTable reloadData];

}


-(IBAction)nearBy
{

    if(dataToShowArray!=nil)
     [dataToShowArray release];

    for(int counter = 0; counter<10; counter++)
    {
     UIView *tempView = [[UIView alloc]
initWithFrame:CGRectMake(0, 0, 320, 135)];
     tempView.backgroundColor = [UIColor blueColor];

     UIImageView *groupImage = [[UIImageView alloc]
initWithFrame:CGRectMake(10, 5, 130, 90)];
     [groupImage setBackgroundColor:[UIColor redColor]];
     groupImage.image = [UIImage
imageNamed:@"backGround.png"];
     [tempView addSubview:groupImage];
     [groupImage release];

     UILabel *venueName= [[UILabel alloc]
initWithFrame:CGRectMake(10, 105, 100, 25)];
     venueName.text = [NSString stringWithFormat: @"Venue
name"];
     venueName.backgroundColor = [UIColor clearColor];
     [tempView addSubview:venueName];
     [venueName release];

     UILabel *address = [[UILabel alloc]
initWithFrame:CGRectMake(175, 5, 80, 125)];
     address.text = [NSString stringWithFormat: @"address
%d",counter];
     address.backgroundColor = [UIColor clearColor];
     address.numberOfLines = 5;
     [tempView addSubview:address];
     [address release];

     UILabel *distance = [[UILabel alloc]
initWithFrame:CGRectMake(275, 65, 100, 25)];
     distance.text = [NSString stringWithFormat: @"distance
%d ",counter];
     distance.backgroundColor = [UIColor clearColor];
     [tempView addSubview:distance];
     [distance release];

     [nearByArray addObject:tempView];
     [tempView release];

    }

    dataToShowArray = [[NSMutableArray alloc]
initWithArray:nearByArray];
    self.addVenueBack.hidden = YES;
    self.venueTable.hidden = NO;

    [self.venueTable reloadData];
}


-(IBAction)addVenue
{
    self.addVenueBack.hidden = NO;
    self.venueTable.hidden = YES;


}
```

```
-(IBAction)addVenueCall
{

}




- (NSInteger)numberOfSectionsInTableView:(UITableView
*)tableView              // Default is 1 if not implemented
{
    return 1;
}

- (NSInteger)tableView:(UITableView *)tableView
numberOfRowsInSection:(NSInteger)section
{
    return [dataToShowArray count];
}

// Row display. Implementers should *always* try to reuse
cells by setting each cell's reuseIdentifier and querying
for available reusable cells with
dequeueReusableCellWithIdentifier:
// Cell gets various attributes set automatically based on
table (separators) and data source (accessory views,
editing controls)

- (UITableViewCell *)tableView:(UITableView *)tableView
cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    //static NSString *CellIdentifier = @"Cell";

    UITableViewCell *cell = [venueTable
dequeueReusableCellWithIdentifier:nil];
    if (cell == nil) {
        cell = [[UITableViewCell alloc]
initWithStyle:UITableViewCellStyleDefault
reuseIdentifier:nil];
    }

    // Configure the cell...
    // cell.textLabel.text = @"Test";
    [cell addSubview:[dataToShowArray
objectAtIndex:indexPath.row]];
    //cell.backgroundColor = [UIColor whiteColor];
    cell.selectionStyle =
UITableViewCellSelectionStyleNone;
    return cell;
}
```

```
- (void)tableView:(UITableView *)tableView
didSelectRowAtIndexPath:(NSIndexPath *)indexPath
{
    VenueDetailsController *VenueDetailsViewControllerObj =
[[VenueDetailsController alloc] init];
    [[self navigationController]
pushViewController:VenueDetailsViewControllerObj
animated:YES];
    [VenueDetailsViewControllerObj release];
}

- (CGFloat)tableView:(VenueDetailsController *)tableView
heightForRowAtIndexPath:(NSIndexPath *)indexPath
{
    return ((UIView*)[dataToShowArray
objectAtIndex:indexPath.row]).frame.size.height;
}

/*
// Override to allow orientations other than the default
portrait orientation.
-
(BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOr
ientation)interfaceOrientation {
    // Return YES for supported orientations.
    return (interfaceOrientation ==
UIInterfaceOrientationPortrait);
}
*/

- (void)didReceiveMemoryWarning {
    // Releases the view if it doesn't have a superview.
    [super didReceiveMemoryWarning];

    // Release any cached data, images, etc. that aren't in
use.
}

- (void)viewDidUnload {
    [super viewDidUnload];
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}


- (void)dealloc {
    [super dealloc];
}
```

```
@end
```

```
//
//  PopularViewController.h
//  GrowndSwell
//
//  Created by ArtiVilla on 4/3/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import <UIKit/UIKit.h>


@interface PopularViewController : UIViewController {

    IBOutlet UIImageView *topBar;
    IBOutlet UISearchBar *searchBar;
    IBOutlet UIButton *tvShowsButton;
    IBOutlet UIButton *moviesButton;
    IBOutlet UIButton *vgButton;
    IBOutlet UITableView *showPopularContents;

    NSMutableArray *dataArray;
    NSMutableArray *viewArray;

}


@property (nonatomic,retain) IBOutlet UIImageView *topBar;
@property (nonatomic,retain) IBOutlet UISearchBar
*searchBar;
@property (nonatomic,retain) IBOutlet UIButton
*tvShowsButton;
@property (nonatomic,retain) IBOutlet UIButton
*moviesButton;
@property (nonatomic,retain) IBOutlet UIButton *vgButton;
@property (nonatomic,retain) IBOutlet UITableView
*showPopularContents;

///Method to update Contents

-(IBAction)updateContents:(id)sender;
@end
```

```
//
//  PopularViewController.m
//  GrowndSwell
//
//  Created by ArtiVilla on 4/3/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import "PopularViewController.h"
#import "PPOfShowsMoviesGame.h"

@implementation PopularViewController
@synthesize
topBar,searchBar,tvShowsButton,moviesButton,vgButton,showPo
pularContents;



// The designated initializer.  Override if you create the
controller programmatically and want to perform
customization that is not appropriate for viewDidLoad.
/*
- (id)initWithNibName:(NSString *)nibNameOrNil
bundle:(NSBundle *)nibBundleOrNil {
    self = [super initWithNibName:nibNameOrNil
bundle:nibBundleOrNil];
    if (self) {
        // Custom initialization.
    }
    return self;
}
*/


// Implement viewDidLoad to do additional setup after
loading the view, typically from a nib.
- (void)viewDidLoad {
    [super viewDidLoad];
    dataArray = [[NSMutableArray alloc] init];
    [dataArray addObject:@"test"];
    [dataArray addObject:@"test"];
    [dataArray addObject:@"test"];
    [dataArray addObject:@"test"];
    [dataArray addObject:@"test"];
    [dataArray addObject:@"test"];
    [dataArray addObject:@"test"];
    [dataArray addObject:@"test"];
    [dataArray addObject:@"test"];
    [dataArray addObject:@"test"];
    [dataArray addObject:@"test"];
    [dataArray addObject:@"test"];
    [self.showPopularContents reloadData];
}


/*
// Override to allow orientations other than the default
portrait orientation.
-
(BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOr
ientation)interfaceOrientation {
    // Return YES for supported orientations.
    return (interfaceOrientation ==
UIInterfaceOrientationPortrait);
}
*/


- (NSInteger)numberOfSectionsInTableView:(UITableView
*)tableView              // Default is 1 if not implemented
{
    return 1;
}

- (NSInteger)tableView:(UITableView *)tableView
numberOfRowsInSection:(NSInteger)section
{
    return [dataArray count];
}

// Row display. Implementers should *always* try to reuse
cells by setting each cell's reuseIdentifier and querying
for available reusable cells with
dequeueReusableCellWithIdentifier:
// Cell gets various attributes set automatically based on
table (separators) and data source (accessory views,
editing controls)

- (UITableViewCell *)tableView:(UITableView *)tableView
cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    //static NSString *CellIdentifier = @"Cell";

    UITableViewCell *cell = [showPopularContents
dequeueReusableCellWithIdentifier:nil];
    if (cell == nil) {
        cell = [[UITableViewCell alloc]
initWithStyle:UITableViewCellStyleDefault
reuseIdentifier:nil];
    }
    // Configure the cell...
     cell.textLabel.text = @"Test";
    //[cell addSubview:[viewArray
objectAtIndex:indexPath.row]];
    //cell.backgroundColor = [UIColor whiteColor];
    cell.selectionStyle =
UITableViewCellSelectionStyleNone;
    return cell;
}


- (void)tableView:(UITableView *)tableView
didSelectRowAtIndexPath:(NSIndexPath *)indexPath
{
    PPOfShowsMoviesGame *PPOfShowsMoviesGameObj =
[[PPOfShowsMoviesGame alloc]
initWithNibName:@"PPOfShowsMoviesGame" bundle:nil];
    [[self navigationController]
pushViewController:PPOfShowsMoviesGameObj animated:YES];
    [PPOfShowsMoviesGameObj release];
}

//
//- (CGFloat)tableView:(UITableView *)tableView
heightForRowAtIndexPath:(NSIndexPath *)indexPath
//{
//  //return ((UIView*)[viewArray
objectAtIndex:indexPath.row]).frame.size.height;
//}


-(IBAction)updateContents:(id)sender
{

}

- (void)didReceiveMemoryWarning {
    // Releases the view if it doesn't have a superview.
    [super didReceiveMemoryWarning];

    // Release any cached data, images, etc. that aren't in
use.
}

- (void)viewDidUnload {
    [super viewDidUnload];
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}
```

```
- (void)dealloc {
    [super dealloc];
}


@end
//
//  PPOfShowsMoviesGame.h
//  GrowndSwell
//
//  Created by ArtiVilla on 4/4/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import <UIKit/UIKit.h>


@interface PPOfShowsMoviesGame : UIViewController {

    IBOutlet UIImageView *topBar;
    IBOutlet UIImageView *profileImage;
    IBOutlet UIImageView *friendsHere;
    IBOutlet UIImageView *friendsHere1;
    IBOutlet UIImageView *friendsHere2;
    IBOutlet UIImageView *friendsHere3;
    IBOutlet UIImageView *friendsHere4;
    IBOutlet UIImageView *friendsHere5;
    IBOutlet UIImageView *friendsHere6;
    IBOutlet UIImageView *friendsHere7;
    IBOutlet UIImageView *friendsHere8;
    IBOutlet UIImageView *friendsHere9;
    IBOutlet UIImageView *friendsHere10;
    IBOutlet UIImageView *friendsHere11;
    IBOutlet UIImageView *friendsHere12;
    IBOutlet UIImageView *friendsHere13;

    IBOutlet UIButton *playButton;
    IBOutlet UIButton *peopleHereButton;

    IBOutlet UILabel *nameOfShow;
    IBOutlet UILabel *currentLable;
    IBOutlet UILabel *detailsOfShow;
    IBOutlet UILabel *peoplePlaying;


}

@property (nonatomic,retain) IBOutlet UIImageView *topBar;
```

```
@property (nonatomic,retain) IBOutlet UIImageView
*profileImage;
@property (nonatomic,retain) IBOutlet UIImageView
*friendsHere;
@property (nonatomic,retain) IBOutlet UIImageView
*friendsHere1;
@property (nonatomic,retain) IBOutlet UIImageView
*friendsHere2;
@property (nonatomic,retain) IBOutlet UIImageView
*friendsHere3;
@property (nonatomic,retain) IBOutlet UIImageView
*friendsHere4;
@property (nonatomic,retain) IBOutlet UIImageView
*friendsHere5;
@property (nonatomic,retain) IBOutlet UIImageView
*friendsHere6;
@property (nonatomic,retain) IBOutlet UIImageView
*friendsHere7;
@property (nonatomic,retain) IBOutlet UIImageView
*friendsHere8;
@property (nonatomic,retain) IBOutlet UIImageView
*friendsHere9;
@property (nonatomic,retain) IBOutlet UIImageView
*friendsHere10;
@property (nonatomic,retain) IBOutlet UIImageView
*friendsHere11;
@property (nonatomic,retain) IBOutlet UIImageView
*friendsHere12;
@property (nonatomic,retain) IBOutlet UIImageView
*friendsHere13;
@property (nonatomic,retain) IBOutlet UIButton *playButton;
@property (nonatomic,retain) IBOutlet UIButton
*peopleHereButton;
@property (nonatomic,retain) IBOutlet UILabel *nameOfShow;
@property (nonatomic,retain) IBOutlet UILabel
*currentLable;
@property (nonatomic,retain) IBOutlet UILabel
*detailsOfShow;
@property (nonatomic,retain) IBOutlet UILabel
*peoplePlaying;


//Methods
-(IBAction)peopleHere:(id)sender;
-(IBAction)playMethod:(id)sender;



@end
```

```
//
//  PPOfShowsMoviesGame.m
//  GrowndSwell
//
//  Created by ArtiVilla on 4/4/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import "PPOfShowsMoviesGame.h"


@implementation PPOfShowsMoviesGame

@synthesize
topBar,profileImage,friendsHere,friendsHere1,friendsHere2,f
riendsHere3,friendsHere4,friendsHere5,friendsHere6,friendsH
ere7,friendsHere8,friendsHere9,friendsHere10,friendsHere11,
friendsHere12,friendsHere13;
@synthesize
playButton,peopleHereButton,nameOfShow,currentLable,details
OfShow,peoplePlaying;
// The designated initializer.  Override if you create the
controller programmatically and want to perform
customization that is not appropriate for viewDidLoad.
/*
- (id)initWithNibName:(NSString *)nibNameOrNil
bundle:(NSBundle *)nibBundleOrNil {
    self = [super initWithNibName:nibNameOrNil
bundle:nibBundleOrNil];
    if (self) {
        // Custom initialization.
    }
    return self;
}
*/

/*
// Implement viewDidLoad to do additional setup after
loading the view, typically from a nib.
- (void)viewDidLoad {
    [super viewDidLoad];
}
*/

/*
// Override to allow orientations other than the default
portrait orientation.
-
(BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOr
ientation)interfaceOrientation {
    // Return YES for supported orientations.
    return (interfaceOrientation ==
UIInterfaceOrientationPortrait);
}
*/

- (void)didReceiveMemoryWarning {
    // Releases the view if it doesn't have a superview.
    [super didReceiveMemoryWarning];

    // Release any cached data, images, etc. that aren't in
use.
}



-(IBAction)peopleHere:(id)sender
{
}
-(IBAction)playMethod:(id)sender
{
}


- (void)viewDidUnload {
    [super viewDidUnload];
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}


- (void)dealloc {
    [super dealloc];
}


@end
```

```
//
//  ProfilePageViewController.h
//  GrowndSwell
//
//  Created by ArtiVilla on 3/29/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import <UIKit/UIKit.h>


@interface ProfilePageViewController : UIViewController {

    IBOutlet UIImageView *profileImage;
    IBOutlet UILabel *friendName;
    IBOutlet UIButton *addButton;

}

@property (nonatomic,retain) IBOutlet UIImageView
*profileImage;
@property (nonatomic,retain) IBOutlet UILabel *friendName;
@property (nonatomic,retain) IBOutlet UIButton *addButton;

-(IBAction) addFriends;
-(IBAction) goBack;


@end
```

```
//
//  ProfilePageViewController.m
//  GrowndSwell
//
//  Created by ArtiVilla on 3/29/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import "ProfilePageViewController.h"


@implementation ProfilePageViewController

@synthesize profileImage,friendName,addButton;
// The designated initializer.  Override if you create the
controller programmatically and want to perform
customization that is not appropriate for viewDidLoad.
/*
- (id)initWithNibName:(NSString *)nibNameOrNil
bundle:(NSBundle *)nibBundleOrNil {
    self = [super initWithNibName:nibNameOrNil
bundle:nibBundleOrNil];
    if (self) {
        // Custom initialization.
    }
    return self;
}
*/


-(IBAction) addFriends
{

}

-(IBAction) goBack
{
    [self.navigationController
popViewControllerAnimated:YES];
}

/*
// Implement viewDidLoad to do additional setup after
loading the view, typically from a nib.
- (void)viewDidLoad {
    [super viewDidLoad];
}
*/

/*
```

```
// Override to allow orientations other than the default
portrait orientation.
-
(BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOr
ientation)interfaceOrientation {
    // Return YES for supported orientations.
    return (interfaceOrientation ==
UIInterfaceOrientationPortrait);
}
*/

- (void)didReceiveMemoryWarning {
    // Releases the view if it doesn't have a superview.
    [super didReceiveMemoryWarning];

    // Release any cached data, images, etc. that aren't in
use.
}

- (void)viewDidUnload {
    [super viewDidUnload];
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}


- (void)dealloc {
    [super dealloc];
}


@end
```

```
//
//  RegisterViewController.h
//  GrowndSwell
//
//  Created by ArtiVilla on 3/18/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import <UIKit/UIKit.h>


@interface RegisterViewController : UIViewController {

}

@end
```

```
//
//  RegisterViewController.m
//  GrowndSwell
//
//  Created by ArtiVilla on 3/18/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import "RegisterViewController.h"


@implementation RegisterViewController

// The designated initializer.  Override if you create the
controller programmatically and want to perform
customization that is not appropriate for viewDidLoad.
/*
- (id)initWithNibName:(NSString *)nibNameOrNil
bundle:(NSBundle *)nibBundleOrNil {
    self = [super initWithNibName:nibNameOrNil
bundle:nibBundleOrNil];
    if (self) {
        // Custom initialization.
    }
    return self;
}
*/


// Implement viewDidLoad to do additional setup after
loading the view, typically from a nib.
- (void)viewDidLoad {
    [super viewDidLoad];
    self.navigationController.navigationBarHidden = NO;
}



/*
// Override to allow orientations other than the default
portrait orientation.
-
(BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOr
ientation)interfaceOrientation {
    // Return YES for supported orientations.
    return (interfaceOrientation ==
UIInterfaceOrientationPortrait);
}
*/

- (void)didReceiveMemoryWarning {
    // Releases the view if it doesn't have a superview.
    [super didReceiveMemoryWarning];

    // Release any cached data, images, etc. that aren't in
use.
}

- (void)viewDidUnload {
    [super viewDidUnload];
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}


- (void)dealloc {
    [super dealloc];
}


@end
```

```
//
//  StudyingAllViewController.h
//  GrowndSwell
//
//  Created by ArtiVilla on 4/3/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import <UIKit/UIKit.h>


@interface StudyingAllViewController : UIViewController {

    IBOutlet UIImageView *topBar;
    IBOutlet UIButton *allButton;
    IBOutlet UIButton *favButton;
    IBOutlet UIButton *addShowButton;
    IBOutlet UITableView *showListTable;
    NSMutableArray *dataArray;
    NSMutableArray *viewArray;

    //Add show View
    IBOutlet UIView *addShowBackView;
    IBOutlet UILabel *showNameLable;
    IBOutlet UILabel *showGenreLable;
    IBOutlet UILabel *showYearLable;

    IBOutlet UITextField *showNameTextField;
    IBOutlet UITextField *showGenreTextField;
    IBOutlet UITextField *showYearTextField;
    IBOutlet UIButton *watchButton;
    //IBOutlet UIButton *watchButton;



}

@property (nonatomic,retain) IBOutlet UIImageView *topBar;
@property (nonatomic,retain) IBOutlet UIButton *allButton;
@property (nonatomic,retain) IBOutlet UIButton *favButton;
@property (nonatomic,retain) IBOutlet UIButton
*addShowButton;
@property (nonatomic,retain) IBOutlet UITableView
*showListTable;

//Add Property
@property (nonatomic,retain) IBOutlet UIView
*addShowBackView;
@property (nonatomic,retain) IBOutlet UILabel
*showNameLable;
```

```
@property (nonatomic,retain) IBOutlet UILabel
*showGenreLable;
@property (nonatomic,retain) IBOutlet UILabel
*showYearLable;
@property (nonatomic,retain) IBOutlet UITextField
*showNameTextField;
@property (nonatomic,retain) IBOutlet UITextField
*showGenreTextField;
@property (nonatomic,retain) IBOutlet UITextField
*showYearTextField;
@property (nonatomic,retain) IBOutlet UIButton
*watchButton;


//Methods
-(IBAction)showdata:(id)sender;
-(void)rowBackView;
-(IBAction)addShow:(id)sender;
-(IBAction)watchShow:(id)sender;

@end
```

```
//
//  StudyingAllViewController.m
//  GrowndSwell
//
//  Created by ArtiVilla on 4/3/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import "StudyingAllViewController.h"


@implementation StudyingAllViewController

@synthesize
topBar,allButton,favButton,addShowButton,showListTable;
@synthesize
watchButton,addShowBackView,showNameLable,showNameTextField
,showGenreLable,showGenreTextField,showYearLable,showYearTe
xtField;
// The designated initializer.  Override if you create the
controller programmatically and want to perform
customization that is not appropriate for viewDidLoad.
/*
- (id)initWithNibName:(NSString *)nibNameOrNil
bundle:(NSBundle *)nibBundleOrNil {
    self = [super initWithNibName:nibNameOrNil
bundle:nibBundleOrNil];
    if (self) {
        // Custom initialization.
    }
    return self;
}
*/


// Implement viewDidLoad to do additional setup after
loading the view, typically from a nib.
- (void)viewDidLoad {
    [super viewDidLoad];
    self.view.backgroundColor = [UIColor redColor];
    dataArray = [[NSMutableArray alloc] init];
    [dataArray addObject:@"test"];
    [dataArray addObject:@"test"];
    [dataArray addObject:@"test"];
    [dataArray addObject:@"test"];
    [dataArray addObject:@"test"];
    [dataArray addObject:@"test"];
    [dataArray addObject:@"test"];
    [dataArray addObject:@"test"];
    [dataArray addObject:@"test"];
    [dataArray addObject:@"test"];
    [dataArray addObject:@"test"];
    [dataArray addObject:@"test"];
    viewArray = [[NSMutableArray alloc] init];
    [self rowBackView];
}


-(IBAction)addShow:(id)sender
{
    self.showListTable.hidden = YES;
    self.addShowBackView.hidden = NO;
    self.showGenreLable.hidden = NO;
    self.showGenreTextField.hidden = NO;
    self.showNameLable.hidden = NO;
    self.showNameTextField.hidden = NO;
    self.showYearLable.hidden = NO;
    self.showYearTextField.hidden = NO;


}

-(void)rowBackView
{
    int totalRows = [dataArray count];
    if([viewArray count]>0)
     [viewArray removeAllObjects];

    //hide add view
    self.addShowBackView.hidden = YES;
    self.showGenreLable.hidden = YES;
    self.showGenreTextField.hidden = YES;
    self.showNameLable.hidden = YES;
    self.showNameTextField.hidden = YES;
    self.showYearLable.hidden = YES;
    self.showYearTextField.hidden = YES;



    self.showListTable.hidden = NO;
    for(int rowCounter = 0; rowCounter
<totalRows;rowCounter++)
    {
     UIView *backView = [[UIView alloc]
initWithFrame:CGRectMake(0, 0, 320, 100)];

     UILabel *nameOfShow = [[UILabel alloc]
initWithFrame:CGRectMake(5, 5, 120, 90)];
     nameOfShow.text = @"Name of show goes here";
     nameOfShow.textAlignment = UITextAlignmentLeft;
```

```
      nameOfShow.backgroundColor = [UIColor clearColor];
      nameOfShow.numberOfLines = 5;
      [backView addSubview:nameOfShow];
      [nameOfShow release];

      UILabel *nameOfdirector = [[UILabel alloc]
initWithFrame:CGRectMake(135, 5, 120, 90)];
      nameOfdirector.text = @"Name of show goes here";
      nameOfdirector.textAlignment = UITextAlignmentLeft;
      nameOfdirector.backgroundColor = [UIColor clearColor];
      nameOfdirector.numberOfLines = 5;
      [backView addSubview:nameOfdirector];
      [nameOfdirector release];

      UIButton *starButton = [UIButton
buttonWithType:UIButtonTypeCustom];
      starButton.frame = CGRectMake(275, 35, 30,30);
      [starButton setImage:[UIImage
imageNamed:@"backGround.png"]
forState:UIControlStateNormal];
      [backView addSubview:starButton];

      [viewArray addObject:backView];
      [backView release];

    }
    [self.showListTable reloadData];
}

-(IBAction)showdata:(id)sender
{
    if([sender tag] == 2)
    {

    }
}


-(IBAction)watchShow:(id)sender
{

}

- (NSInteger)numberOfSectionsInTableView:(UITableView
*)tableView              // Default is 1 if not implemented
{
    return 1;
}

- (NSInteger)tableView:(UITableView *)tableView
```

```
numberOfRowsInSection:(NSInteger)section
{
    return [viewArray count];
}

// Row display. Implementers should *always* try to reuse
cells by setting each cell's reuseIdentifier and querying
for available reusable cells with
dequeueReusableCellWithIdentifier:
// Cell gets various attributes set automatically based on
table (separators) and data source (accessory views,
editing controls)

- (UITableViewCell *)tableView:(UITableView *)tableView
cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    //static NSString *CellIdentifier = @"Cell";

    UITableViewCell *cell = [showListTable
dequeueReusableCellWithIdentifier:nil];
    if (cell == nil) {
        cell = [[UITableViewCell alloc]
initWithStyle:UITableViewCellStyleDefault
reuseIdentifier:nil];
    }

    // Configure the cell...
    // cell.textLabel.text = @"Test";
    [cell addSubview:[viewArray
objectAtIndex:indexPath.row]];
    //cell.backgroundColor = [UIColor whiteColor];
    cell.selectionStyle =
UITableViewCellSelectionStyleNone;
    return cell;
}


- (CGFloat)tableView:(UITableView *)tableView
heightForRowAtIndexPath:(NSIndexPath *)indexPath
{
    return ((UIView*)[viewArray
objectAtIndex:indexPath.row]).frame.size.height;
}


/*
// Override to allow orientations other than the default
portrait orientation.
-
(BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOr
ientation)interfaceOrientation {
    // Return YES for supported orientations.
    return (interfaceOrientation ==
UIInterfaceOrientationPortrait);
}
*/

- (void)didReceiveMemoryWarning {
    // Releases the view if it doesn't have a superview.
    [super didReceiveMemoryWarning];

    // Release any cached data, images, etc. that aren't in
use.
}

- (void)viewDidUnload {
    [super viewDidUnload];
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}


- (void)dealloc {
    [super dealloc];
}


@end
```

```
//
//  SwellsViewController.h
//  GrowndSwell
//
//  Created by ArtiVilla on 3/26/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import <UIKit/UIKit.h>


@interface SwellsViewController : UIViewController {

}

-(id) initWithTabBar;

@end
```

```
//
//  SwellsViewController.m
//  GrowndSwell
//
//  Created by ArtiVilla on 3/26/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import "SwellsViewController.h"


@implementation SwellsViewController

-(id) initWithTabBar
{
    self = [super init];
    if (self)
    {
     //this is the label on the tab button itself
     self.title = @"Swells";
     //parsingType = [[NSString
alloc]initWithString:@"recent"];
     //Image of the Tab
     //loadStatus = @"loading";
     self.tabBarItem.image = [UIImage
imageNamed:@"Places.png"];
    }
    return self;
}

/*
// Implement viewDidLoad to do additional setup after
loading the view, typically from a nib.
- (void)viewDidLoad {
    [super viewDidLoad];
}
*/

/*
// Override to allow orientations other than the default
portrait orientation.
-
(BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOr
ientation)interfaceOrientation {
    // Return YES for supported orientations.
    return (interfaceOrientation ==
UIInterfaceOrientationPortrait);
}
*/
```

```
- (void)didReceiveMemoryWarning {
    // Releases the view if it doesn't have a superview.
    [super didReceiveMemoryWarning];

    // Release any cached data, images, etc. that aren't in
use.
}

- (void)viewDidUnload {
    [super viewDidUnload];
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}


- (void)dealloc {
    [super dealloc];
}


@end
```

```
//
//  VenueDetailsController.h
//  GrowndSwell
//
//  Created by ArtiVilla on 3/20/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import <UIKit/UIKit.h>


@interface VenueDetailsController : UIViewController {

    IBOutlet UIImageView *topBarImage;
    IBOutlet UIImageView *venueImage;

    IBOutlet UILabel *venuName;
    IBOutlet UILabel *venueAddress;

    IBOutlet UIButton *checInButton;
    IBOutlet UIButton *addToAgenda;

}


@property (nonatomic,retain) IBOutlet UIImageView
*venueImage;
@property (nonatomic,retain) IBOutlet UIImageView
*topBarImage;
@property (nonatomic,retain) IBOutlet UILabel *venuName;
@property (nonatomic,retain) IBOutlet  UILabel
*venueAddress;
@property (nonatomic,retain) IBOutlet UIButton
*checInButton;
@property (nonatomic,retain) IBOutlet UIButton
*addToAgenda;


-(IBAction) addToAgenda;
-(IBAction) goBack;
-(IBAction) peopleHere;
-(IBAction) checkInMethod;

@end
```

```
//
//  VenueDetailsController.m
//  GrowndSwell
//
//  Created by ArtiVilla on 3/20/11.
//  Copyright 2011 __GrowndSwell__. All rights reserved.
//

#import "VenueDetailsController.h"
#import "PeopleHereViewController.h"


@implementation VenueDetailsController
@synthesize
topBarImage,venueImage,venuName,venueAddress,checInButton,a
ddToAgenda;

// The designated initializer.  Override if you create the
controller programmatically and want to perform
customization that is not appropriate for viewDidLoad.
/*
- (id)initWithNibName:(NSString *)nibNameOrNil
bundle:(NSBundle *)nibBundleOrNil {
    self = [super initWithNibName:nibNameOrNil
bundle:nibBundleOrNil];
    if (self) {
        // Custom initialization.
    }
    return self;
}
*/

/*
// Implement viewDidLoad to do additional setup after
loading the view, typically from a nib.
- (void)viewDidLoad {
    [super viewDidLoad];
}
*/

/*
// Override to allow orientations other than the default
portrait orientation.
-
(BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOr
ientation)interfaceOrientation {
    // Return YES for supported orientations.
    return (interfaceOrientation ==
UIInterfaceOrientationPortrait);
}
```

```
*/


-(IBAction) addToAgenda
{
}
-(IBAction) checkInMethod
{
}


-(IBAction) peopleHere
{
    PeopleHereViewController *PeopleHereViewControllerObj =
[[PeopleHereViewController alloc] init];
    [self.navigationController
pushViewController:PeopleHereViewControllerObj
animated:YES];
    [PeopleHereViewControllerObj release];
}

-(IBAction)goBack
{
    [self.navigationController
popViewControllerAnimated:YES];
}


- (void)didReceiveMemoryWarning {
    // Releases the view if it doesn't have a superview.
    [super didReceiveMemoryWarning];

    // Release any cached data, images, etc. that aren't in
use.
}

- (void)viewDidUnload {
    [super viewDidUnload];
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}


- (void)dealloc {
    [super dealloc];
}


@end
```

Sources Cited and Consulted

Business Plan Sources Cited and Consulted:

Anonymous. "Android Soars, but IPhone Still Most Desired as Smartphones Grab 25% of U.S. Mobile Market | Nielsen Wire." *Nielsen.com*. The Nielsen Company, 2 Aug. 2010. Web. 21 Mar. 2011. <http://blog.nielsen.com/nielsenwire/online_mobile/android-soars-but-iphone-still-most-desired-as-smartphones-grab-25-of-u-s-mobile-market/>.

"Global Faces and Networked Places A Nielsen report on Social Networking's New Global Footprint." *Nielsen* (2009): 4-12. Web. 20 Mar 2011.

Hazlett, Bob. "Social Networking Statistics And Trends." *Slideshare.com*. OneHalfAmazing.com. Web. 21 Mar. 2011. <http://www.slideshare.net/onehalfamazing/social-networking-statistics-and-trends-presentation>.

"IBISWorld Industry Report 51913b Internet Publishing and Broadcasting in the US." *IBISWorld*. N.p., n.d. Web. 20 Mar 2011. http://www.ibisworld.com.libezproxy2.syr.edu/industryus/default.aspx?indid=1974

Kessler, Scott. "Computers: Consumer Services & the Internet." *Standards & Poor's*. N.p., 14 Oct 2010. Web. 20 Mar 2011. http://www.netadvantage.standardandpoors.com.libezproxy2.syr.edu/NASApp/NetAdvantage/showIndustrySurvey.do?code=csi

Online Podcasts:

Cannistaro, Alan & Shaffer, Josh. 2010. iPhone Application Development. Podcast retrieved from Stanford on iTunes U: http://itunes.stanford.edu/index.html

Book:

Conway, Joe & Hillegass, Aaron. (2010). iPhone Programming, THE BIG NERD RANCH GUIDE. Atlanta, GA: Pearson Technology Group.

Online Forums

Mac Rumors:

http://forums.macrumors.com/forumdisplay.php?f=135

Apple Developer Forums:

http://developer.apple.com/devforums/

Apple documentation

http://developer.apple.com/devcenter/ios/index.action

Facebook API

http://developers.facebook.com/

Twitter API:

http://dev.twitter.com/doc

Google Maps API

http://code.google.com/apis/libraries/devguide.html

APPENDIX A.

GrowndSwell Financial Projections

INCOME STATEMENT

	Summer	**Summer**	**Summer**													
Dates	**6/1/2011**	**7/1/2011**	**8/1/2011**	**9/1/2011**	**10/1/2011**	**11/1/2011**	**12/1/2011**	**1/1/2012**	**2/1/2012**	**3/1/2012**	**4/1/2012**	**5/1/2012**				
	Month 1	**Month 2**	**Month 3**	**Month 4**	**Month 5**	**Month 6**	**Month 7**	**Month 8**	**Month 9**	**Month 10**	**Month 11**	**Month 12**	**Year 2**	**Year 3**	**Year 4**	**Year 5**
Sales Revenue	$0	$0	$0	$0	$1,400	$2,800	$4,200	$5,600	$7,000	$9,800	$11,200	$12,600	$336,000	$504,000	$672,000	$840,000
Fixed Costs																
Wage Expenses	$6,666	$6,666	$6,666	$9,666	$9,666	$12,666	$12,666	$12,666	$15,666	$15,666	$15,666	$15,666	$281,988	$422,982	$634,473	$951,710
Office Supplies	$300	$300	$300	$300	$300	$300	$300	$300	$300	$300	$300	$300	$5,400	$8,100	$12,150	$18,225
Office Lease	$1,000	$1,000	$1,000	$1,300	$1,300	$1,300	$1,300	$1,300	$1,300	$1,300	$1,300	$1,300	$24,000	$36,000	$54,000	$81,000
Founder Expenses	$8,000	$8,000	$8,000	$4,000	$4,000	$4,000	$4,000	$4,000	$4,000	$4,000	$4,000	$4,000	$96,000	$96,000	$96,000	$96,000
Earning	$5,000	$5,000	$5,000	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0
Total Fixed Costs	$10,966	$10,966	$10,966	$15,266	$15,266	$18,266	$18,266	$18,266	$21,266	$21,266	$21,266	$21,266	$407,388	$563,082	$796,623	$1,146,935
Variable Costs																
Start Up Costs																
Computer and Other Equipment	$3,100	$0	$0	$400	$0	$400	$0	$0	$0	$0	$0	$0	$1,000	$1,000	$1,000	$1,000
Servers	$0	$0	$0	$100	$100	$100	$100	$100	$100	$100	$100	$100	$1,800	$2,700	$4,050	$6,075
Growndswell Incorporation and Logo Registration	$700	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0
Other (App store registration)	$100	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$100	$100	$100	$100
Security Deposit	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0
Travel	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0
Advertising, Merchandising, and Promotions**	$3,000	$3,000	$3,000	$3,000	$3,000	$3,000	$3,000	$3,000	$3,000	$3,000	$3,000	$3,000	$64,000	$96,000	$144,000	$216,000
Total Variable Costs	$6,900	$3,000	$3,000	$3,500	$3,100	$3,500	$3,100	$3,100	$3,100	$3,100	$3,100	$3,100	$66,900	$99,800	$149,150	$223,175
Opearting Income	-$17,866	-$13,966	-$13,966	-$18,766	-$16,966	-$18,966	-$17,166	-$15,766	-$17,366	-$14,566	-$13,166	-$11,766	-$138,288	-$158,882	-$273,773	-$530,110

**2 Founders will be responsible for social media marketing so no interns hired initially
First year Advertising, Merchandising, and Promotion expenses sum to $34,000 and scale by 1.5 thereafter

Growndswell Sales Projection Research
Foursquare (a major competitor) has an average 2M Check-In/Week.
Several Assumptions made:
Estimated a growth rate of Check-In/Week rate to increase by 1M/Year which is a conservative statistic considering the boom in the social media industry.
This revenue is only generated from our Going-Out feature and excludes revenue statistics from the Studying and Staying-In features online order systems, advertisements, virtual goods, and distributing user information

Dates	**Month 5**	**Month 6**	**Month 7**	**Month 8**	**Month 9**	**Month 10**	**Month 11**	**Month 12**	**Year 2**	**Year 3**	**Year 4**	**Year 5**
# Check-In/Week	100000	200000	300000	400000	500000	700000	800000	900000	2000000	3000000	4000000	5000000
# Check-In/Month	400000	800000	1200000	1600000	2000000	2800000	3200000	3600000	8000000	12000000	16000000	20000000
% of users referred by Growndswell	2%	2%	2%	2%	2%	2%	2%	2%	2%	2%	2%	2%
# of users referred by Growndswell/Month	8000	16000	24000	32000	40000	56000	64000	72000	160000	240000	320000	400000
Avg. % of users who spend in store	25%	25%	25%	25%	25%	25%	25%	25%	25%	25%	25%	25%
# users who spend in store	2000	4000	6000	8000	10000	14000	16000	18000	40000	60000	80000	100000
Avg. $ Spent in Store/User	$7	$7	$7	$7	$7	$7	$7	$7	$7	$7	$7	$7
$ earned by Store/Month	$14,000	$28,000	$42,000	$56,000	$70,000	$98,000	$112,000	$126,000	$280,000	$420,000	$560,000	$700,000
Growndswell Fee	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%
Growndswell Earnings/Month	$1,400	$2,800	$4,200	$5,600	$7,000	$9,800	$11,200	$12,600	$28,000	$42,000	$56,000	$70,000
Growndswell Earnings/Year									$336,000	$504,000	$672,000	$840,000

http://techcrunch.com/2010/12/08/foursquare-hits-2-million-check-ins-25k-new-users-daily/

Wage Expenses

India Office	**1 Month/Employee**	**1 Year/Employee**	**# employees**	**1 Month**	**1 Year**
Project Manager	$2,222	$26,664	1	$2,222	$26,664
2 Web Developers	$1,111	$13,332	2	$2,222	$26,664
2 Phone Developers	$1,111	$13,332	2	$2,222	$26,664
US Office					
1 Web Developer*	$3,000	$36,000	1	$3,000	$36,000
1 Phone Developer*	$3,000	$36,000	1	$3,000	$36,000
Graphic Designer	$3,000	$36,000	1	$3,000	$36,000
Total				$12,666	$151,992

*Provide minimum wages with 2-3% stock option

http://www.elitefreelancing.com/business/money/average-salary-for-web-designer-web-developer-jobs/
http://www.indeed.com/salary/q-Mobile-Application-Developer-l-United-States.html

Summary

The Concept

Growndswell is a location based social networking service that focuses on the organization of people rather than organization of places. GrowndSwell is an itinerary-based app that helps users efficiently create their social agendas that can be published to your friends network. Setting up your status brings users the flexibility to attend and share new events. Popular social agendas once posted are visible to friends. The platform encompasses all aspects of a college student's social life including going to bars, restaurants, and clubs; staying in to watch movies and shows or play games; and study with friends or classmates. The simplicity of the three features allows a user to view and attend trending venues, watch movies and shows with friends, and find study buddies in their vicinity. Our phone application will organize social activity in real time allowing users to make plans for a more social evening. We also take a group pricing approach to monetizing our business. Through our website, businesses can set up incentives to entice users to come in and spend their evening at their locations.

Monetization

GrowndSwell will provide the businesses with a platform to effectively manage their current and prospective customers. GrowndSwell not only provides businesses with the ability to offer 'Swells' that entice users to frequently visit their stores and bars, but also constructs an effective medium to attract network of friends of existing users.

Unlocking friend discounts through social network is effective because it creates value for customers with group pricing but at the same time brings awareness of new products and promotions.

GrowndSwell can charge a periodic subscription service or a pay-per-use to every ‘Swell’ broadcasted from business benefiting from the service. Our initial service will be free of cost to evaluate realistic potential margins. We will provide an easy-to-navigate web service to provide user trends and other analytics, monitor retention rates and potential invitation leads. For businesses, having a list of prospective customers and user demographics prior to the event has infinite benefits.

Competitive Advantage

Founded by a student start-up, our competitive advantage is our ability to tap into our existing networks. Our initial plan is to market GrowndSwell to college campuses; however, we feel that this service is applicable to people of all ages.

Our marketing strategy is to capitalize and successfully disseminate our phone application to prospective customers through already existing and established social networking media such as Facebook and Twitter. We plan on employing various representatives from different college campuses to help broadcast our product effectively across the nation. We anticipate an upsurge in brand awareness through word of mouth to reach our ideal audience goal.

Funding

GrowndSwell would require initial incubator for the proof of concept and to gain traction to attain a large network of a friends' user base. After the initial alpha-version, it would require a second stage of seed funding in order to hire a full team of in-house engineers to expand to several smartphone platforms. Our initial costs can come around to $15,000 for incubator housing and another $50,000-$100,000 for our seed funding.

The GrowndSwell Team

We are a student startup company that consists of four passionate students from a variety of majors including finance, marketing, engineering and information science with several holding second degrees in entrepreneurship. Our team manifests a strong work ethic, which propels us to move quickly and proficiently at obtaining company objectives necessary for progression. Lastly, we have the ability to think forward and alleviate any anticipated or incurred complications that may come our way. Our team's ability to execute aggressive marketing strategies to college demographics while maintaining college status is a huge competitive advantage; we can effectively tap into our community's ambassadors through broadcasting events. Not many founders are able to tap into this market as they start to gauge interest following their graduation.

Our solution is simple and effective to solve the existing issues arising around not knowing our friends social agendas.

Our team manifests a strong blend of unique skills; from understanding dynamic product engineering practices to business development and generating effective sales leads. This is accomplished through the diversity of our education and the resources each of us bring to the table. Having access to set-up a low cost development team through hiring full-time in-house employees in India can drastically reduce our start-up costs. With advisors and mentors from Silicon Valley having the domain skills to help us through the expansion process, we are an excellent team to depend on. Ultimately, along with passion and commitment, our strong support group and resources is why we should be relied upon to make this vision a success.

Focus of the Capstone Project

The capstone project was focused towards the software engineering development lifecycle that was used to implement the business into a tangible reality. The documentation created to organize our ideas into this tangible form will also be used by software developers and project managers to understand and enhance designs and features in the phone application.

The original source code in this application is the individual effort of the author of the capstone project. The other design and technical documentation including flowcharts and function documents are also contributions of the author of the capstone project.

A waterfall model was used in the software engineering approach for this capstone from requirements, design, implementation and testing.

Significance of Project

The project was critical to my self-development and understanding of software engineering practices for larger projects, to help towards progression in my career as a business analyst and lead engineer for other similar technology start-up companies.

Having the opportunity to work with a team of enthusiastic and bright team members towards a vision, while strengthening my technical skills has certainly influenced my decision to devote full time commitment to this project.