SOCIALIZE GPS

Dylan Kehres

Michael Schott

Table of Contents:

Introduction:

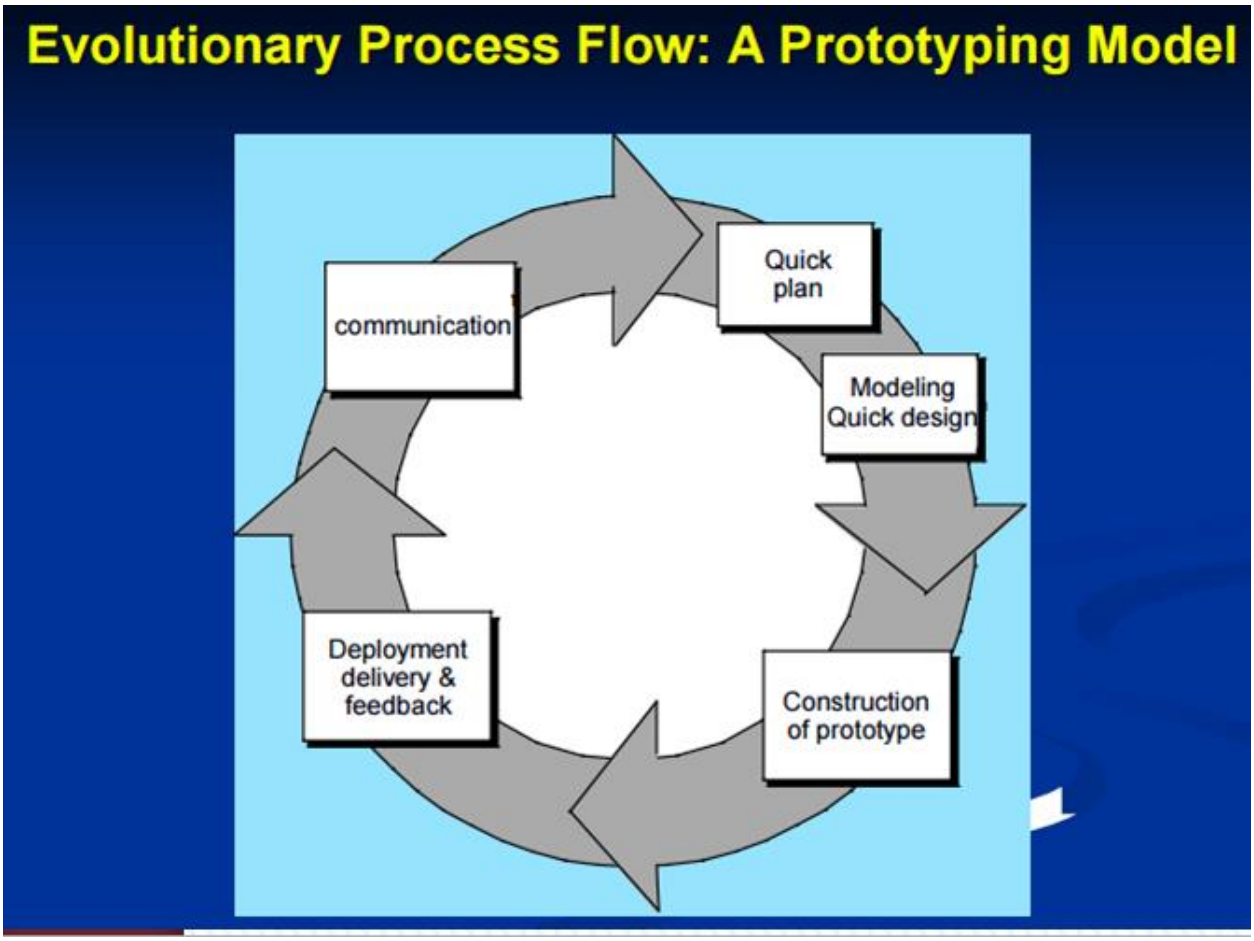
The task at hand is to create a GPS software application for mobile devices that integrates aspects of social media and promotes use through gamification. However the primary use of the application should prioritize supporting navigation above all else.

Narrative:

This software application should be supported on mobile operating systems, beginning with iOS and Android and, should funds permit, other mobile operating systems as well. There is no need for this application to be integrated with PC operating systems such as Windows. The application should have internet capabilities and must have access to the User's location in order to operate. It is to be written in JavaScript.

As stated previously, the main function is Navigation; followed by being able to view maps, find locations, adding and removing friends, leaving feedback at visited locations, challenging and adding rivals (users with similar scores in local proximity), saving frequented locations, viewing leaderboards, obtaining achievements, editing account settings, and viewing friends' locations on a map.

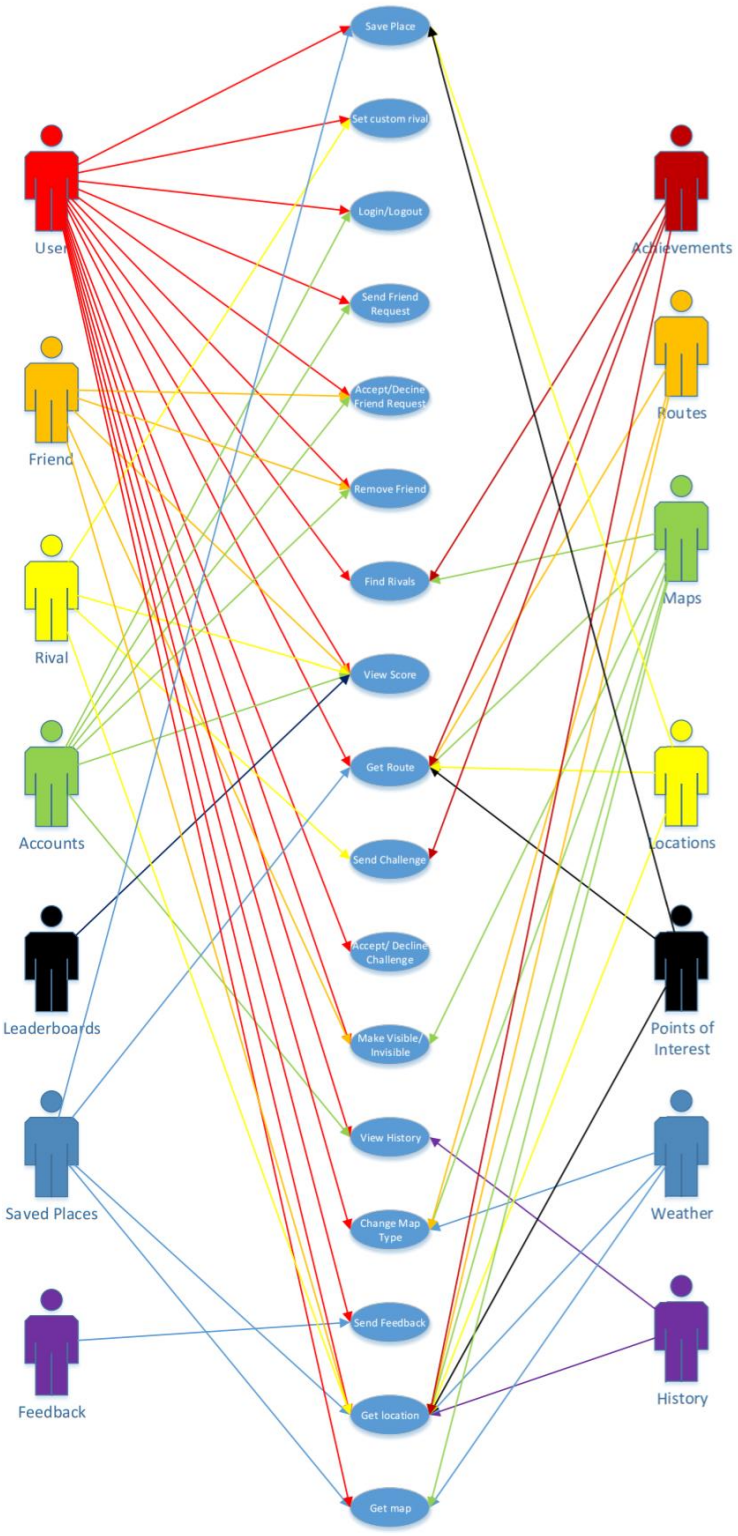
The system will require internet connectivity and the mobile devices using the system must have a GPS unit. This project is estimated to take between 112 and 124 man months. The budget provided is set at $1,000,000.00.

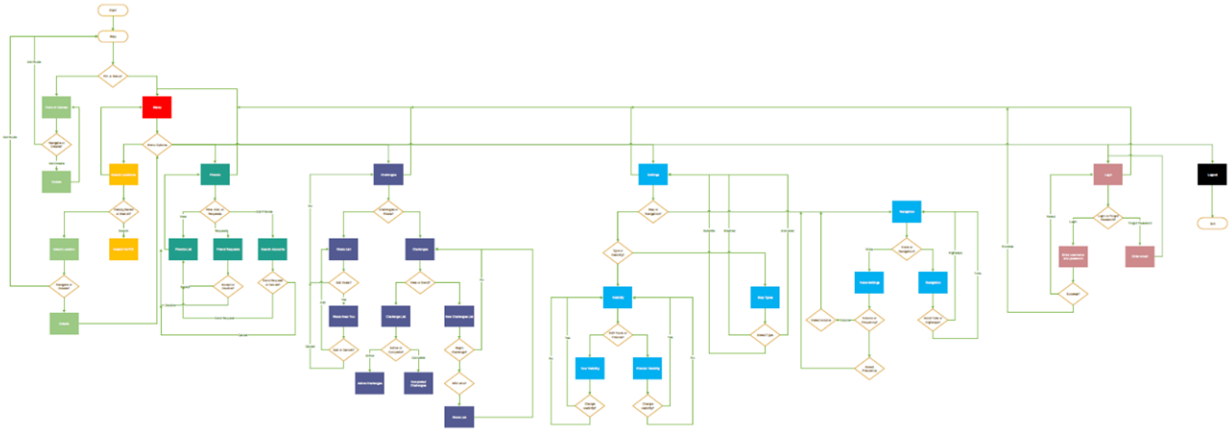


Software Engineering Prescriptive Model:

- Our software is blending social media and GPS navigation with a gaming aspect; therefore it is critical that the User is enjoying the product.
- Users will be able to provide us with important feedback on aspects of our application such as the GUI; which is critical as our application must be user friendly.
- Users will also be able to let us know what features need to be added in future updates of the application that will enhance the User's experience.

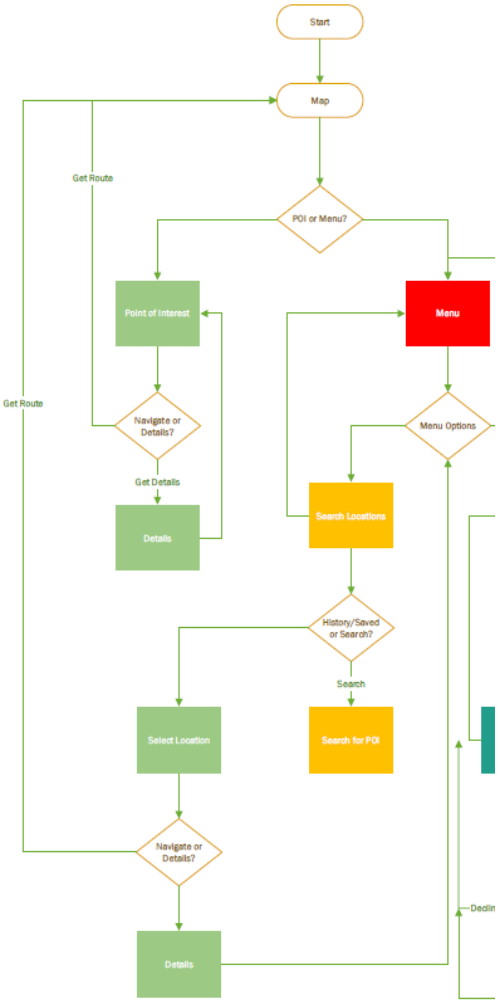
Use Case Diagram:


Save Place
Set custom rival
Login/Logout
Send Friend Request
Accept/Decine Friend Request
Remove Friend
Find Rivals
View Score
Get Route
Send Challenge
Accept/ Decline Challenge
Make Visible/ Invisible
View History
Change Map Type
Send Feedback
Get location
Get map
User
Friend
Rival
Accounts
Leaderboards
Saved Places
Feedback
Achievements
Routes
Maps
Locations
Points of Interest
Weather
History

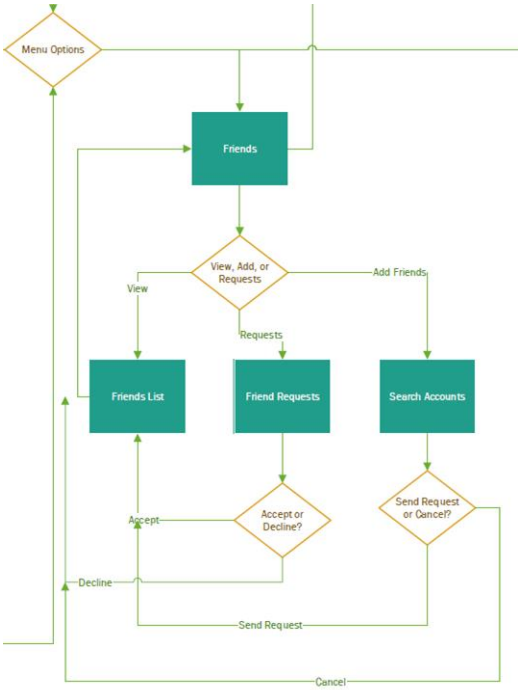


Activity Diagram:

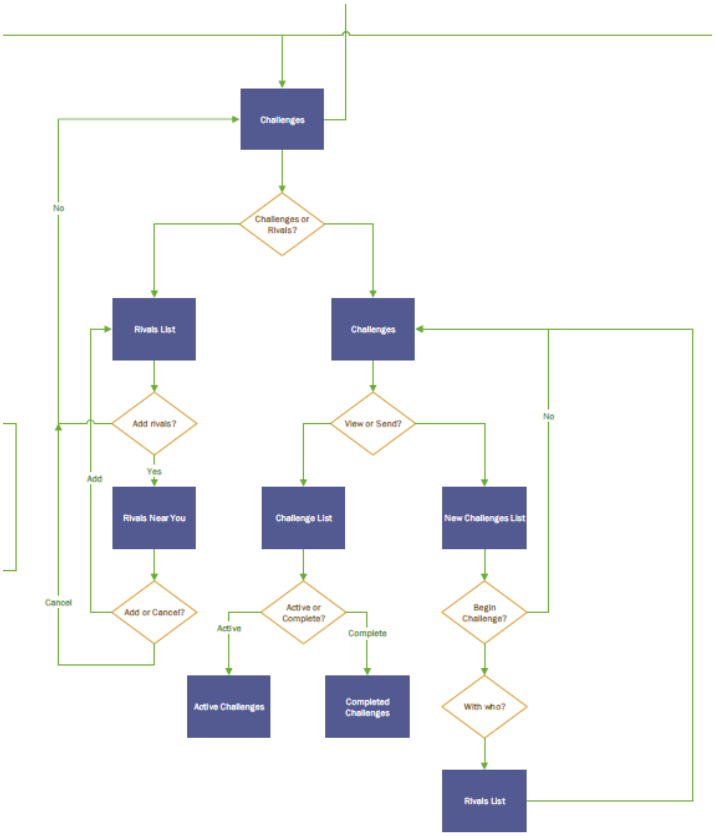
Points of interest and Menu:


Start
Map
Get Route
POI or Menu?
Point of Interest
Menu
Get Route
Navigate or Details?
Menu Options
Get Details
Details
Search Locations
History/Saved or Search?
Search
Select Location
Search for POI
Navigate or Details?
Details

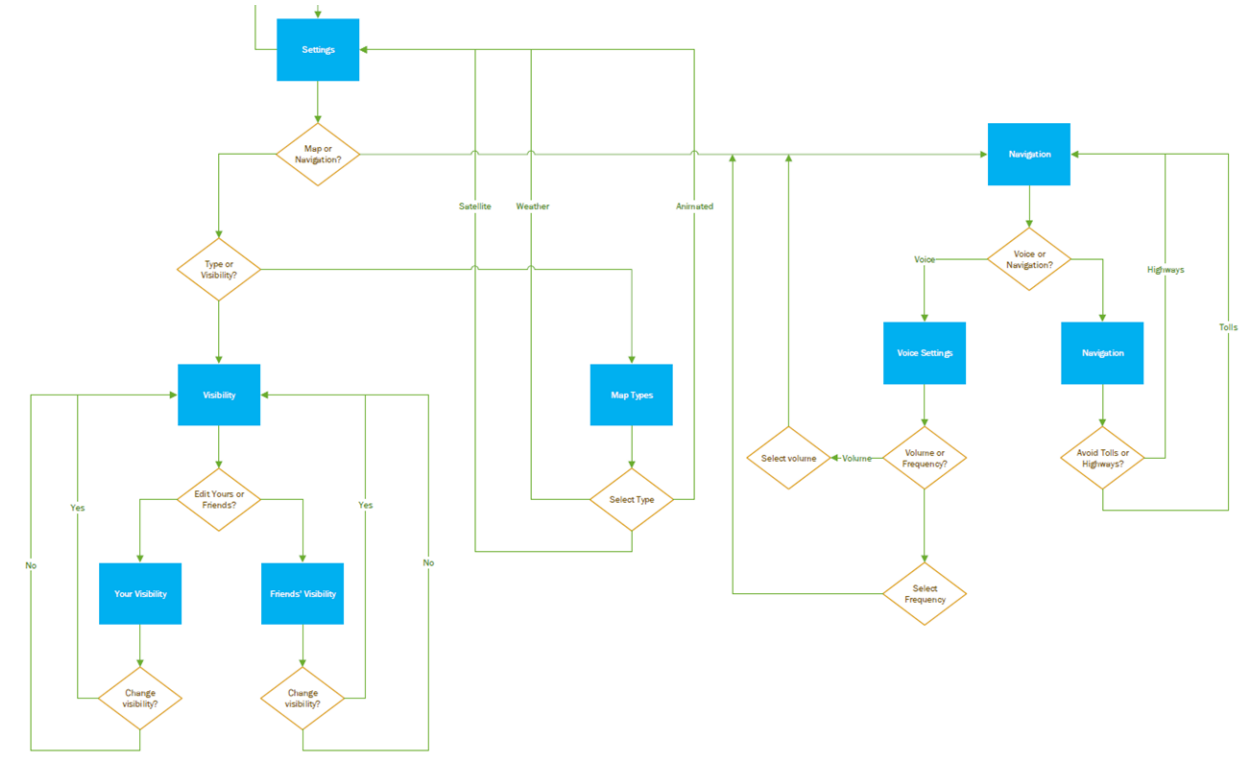
Friends:



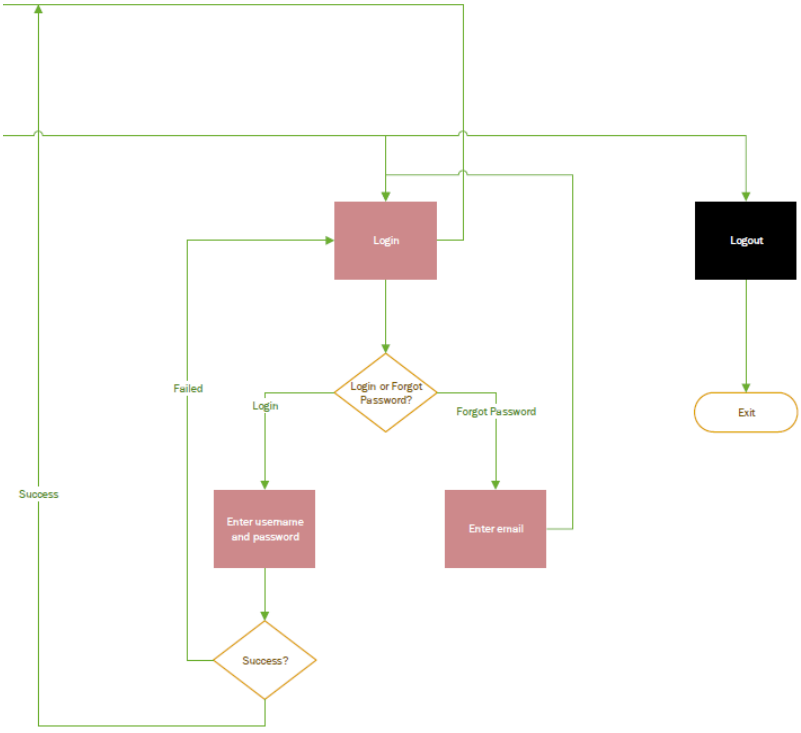
Challenges:



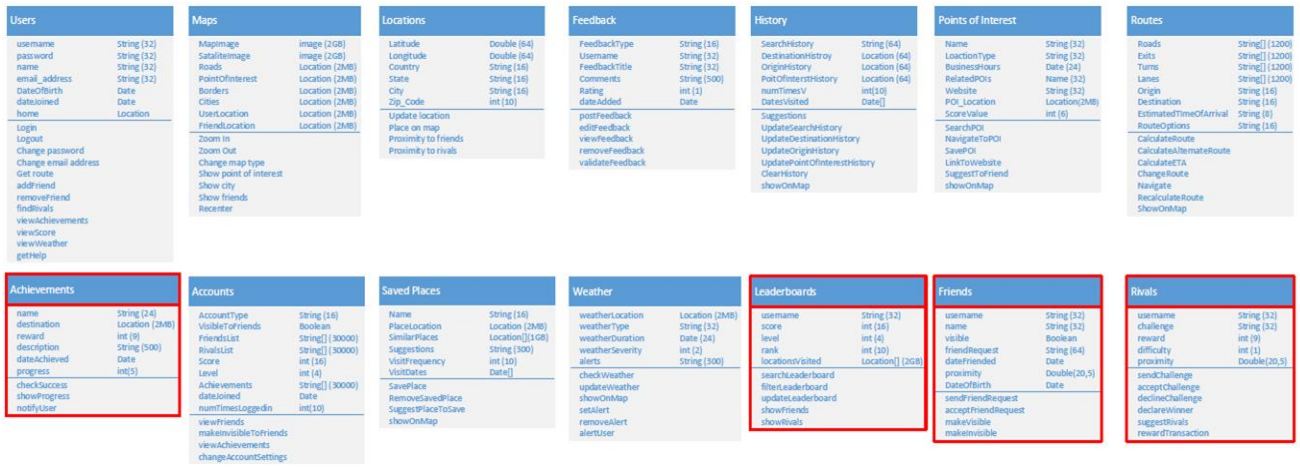
Settings:



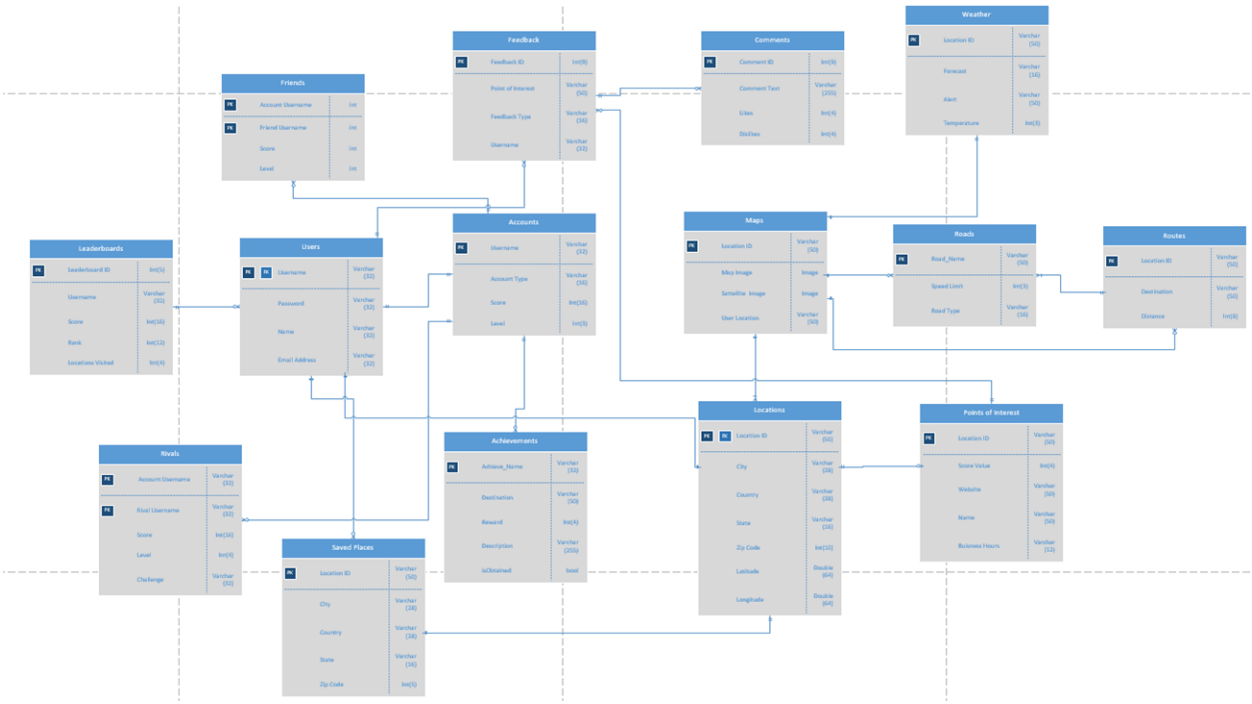
Login and Logout:



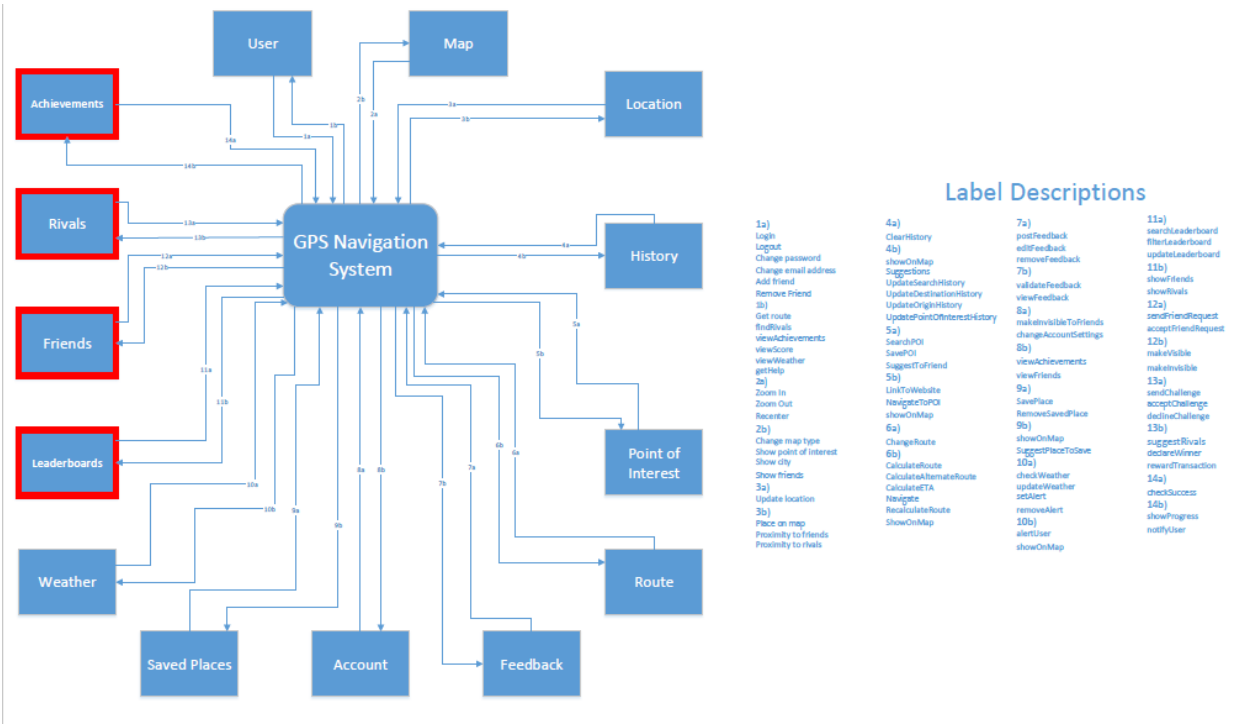
Class Object Diagram:



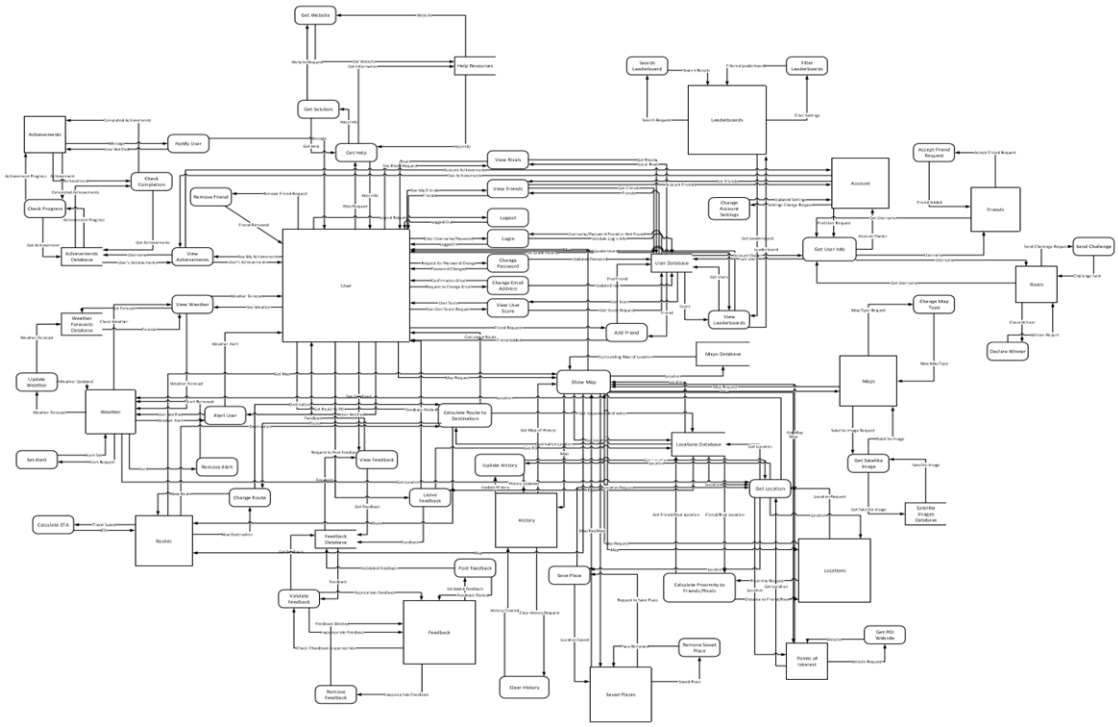
Optional Entity Relationship Diagram:



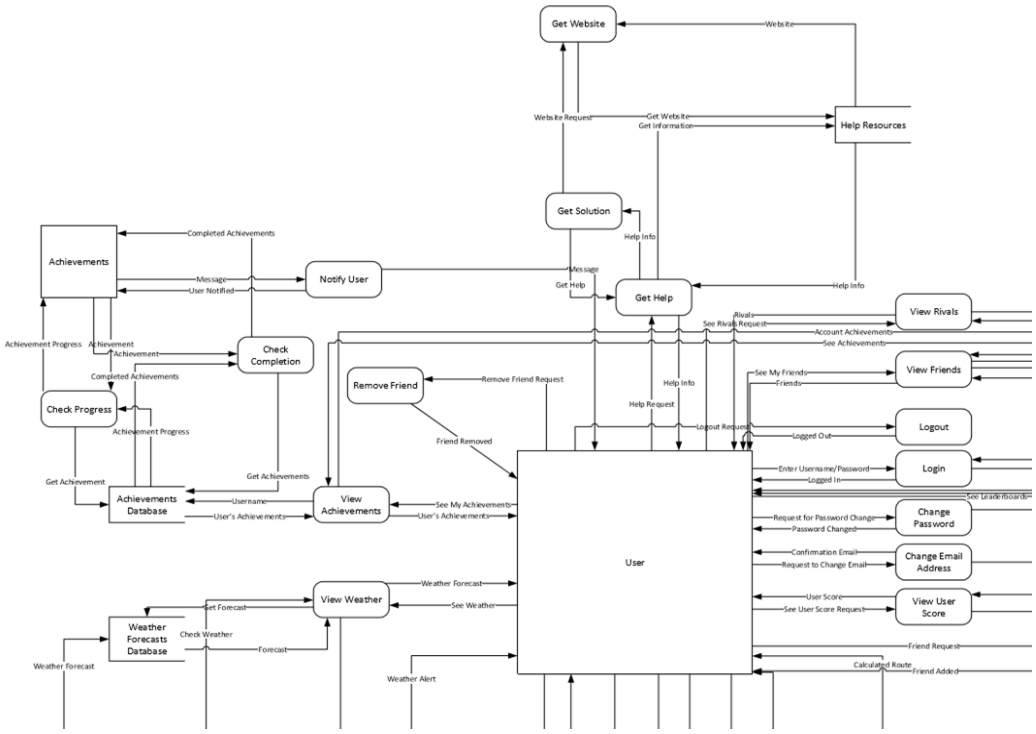
Context Data Flow Diagram:



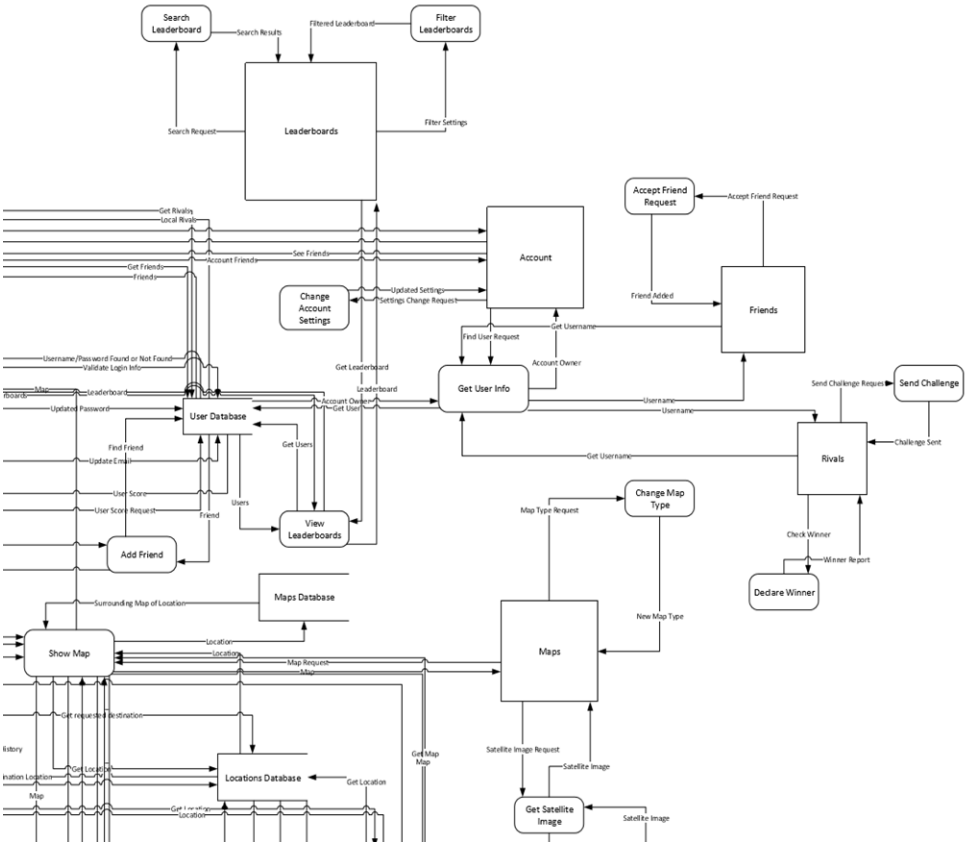
Level-0 Data Flow Diagram:



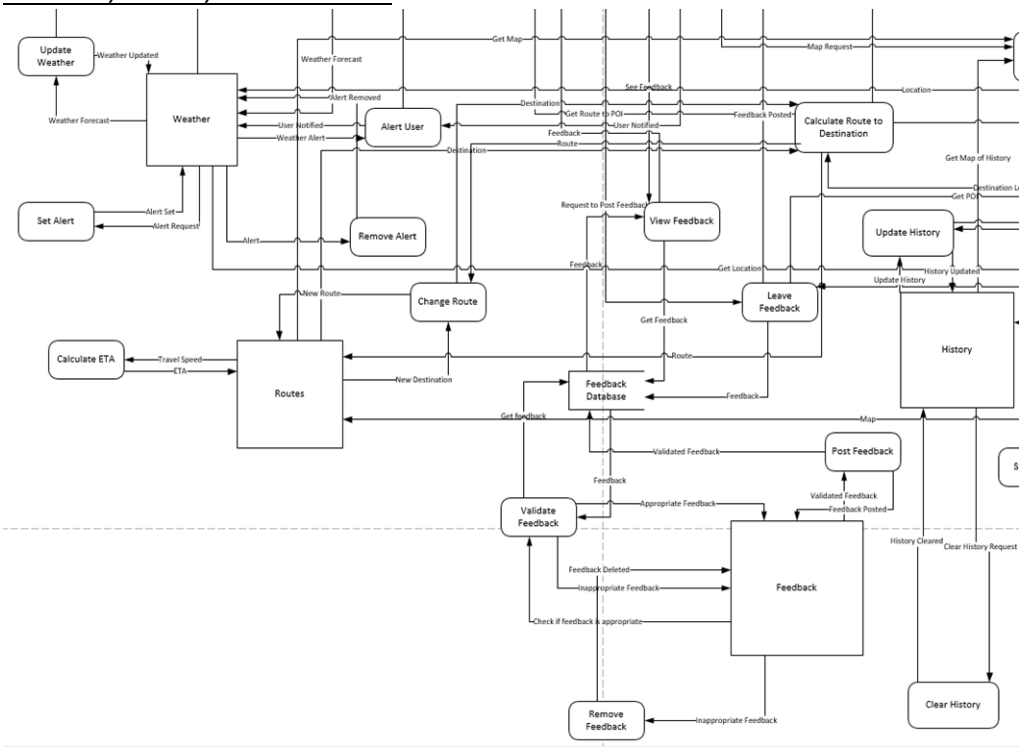
User and Achievements COs:



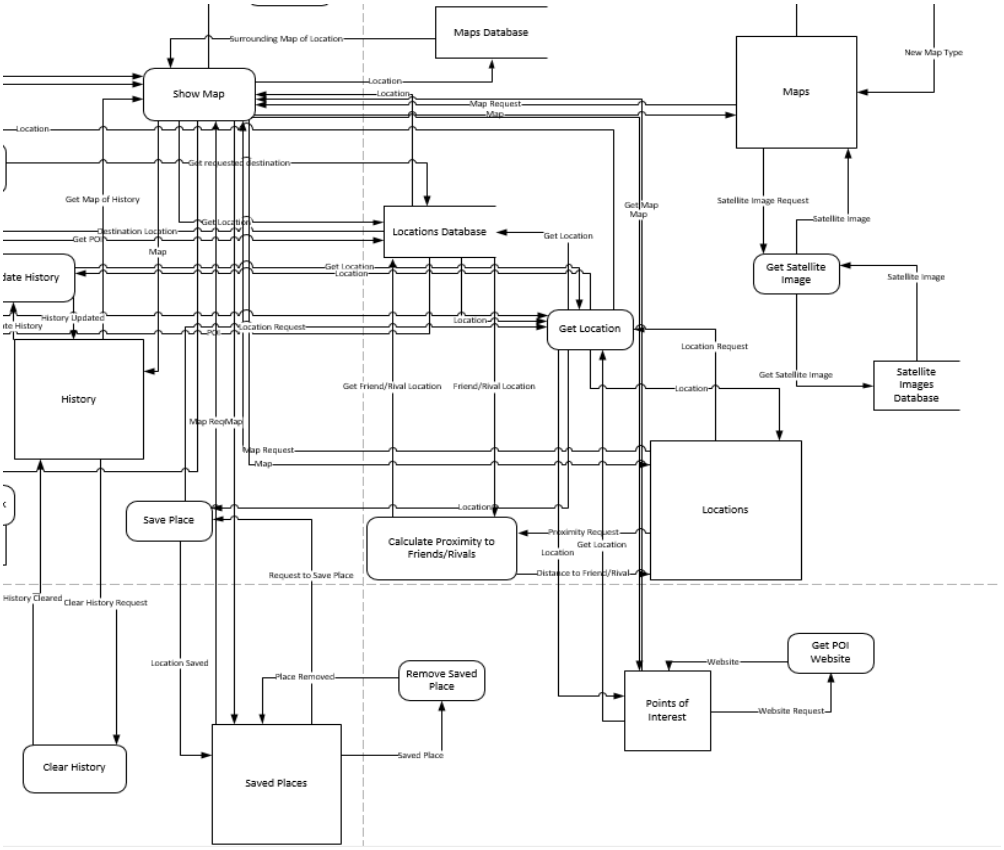
Leaderboards, Account, Maps:



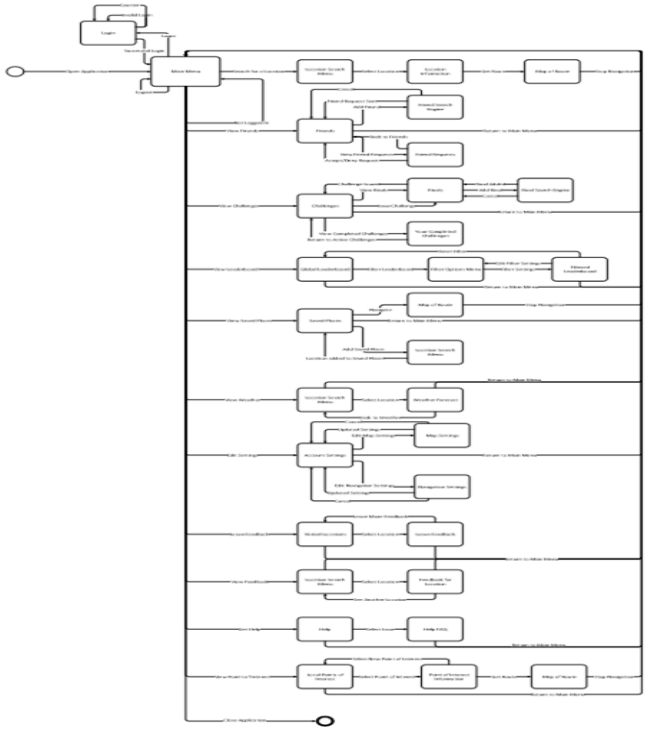
Weather, Routes, and Feedback:



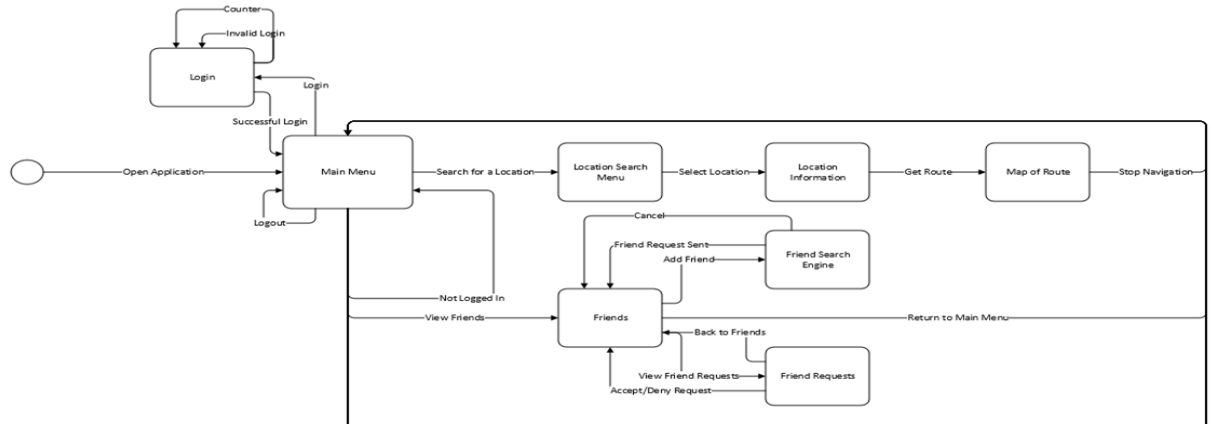
History, Saved Places, Locations:



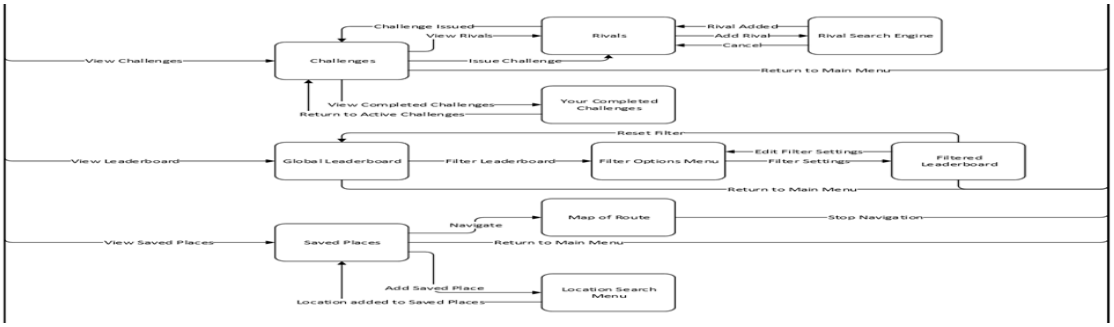
State Transition Diagram:



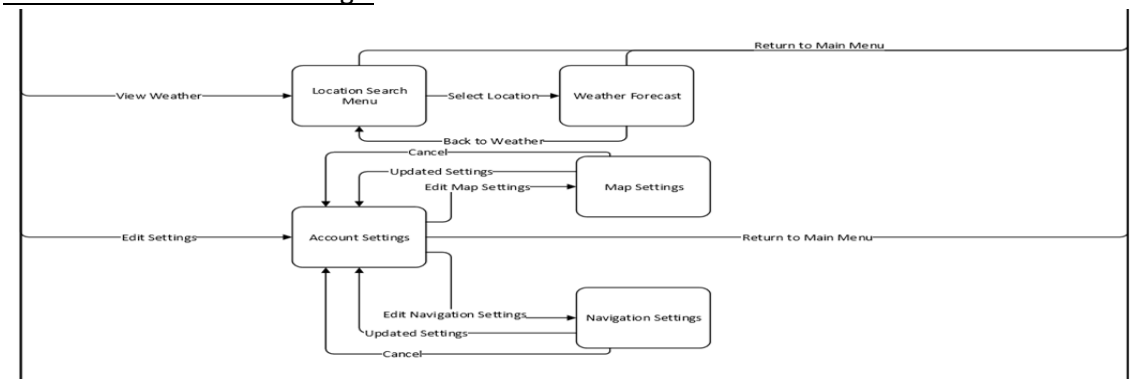
Login, Friends, and Locations:



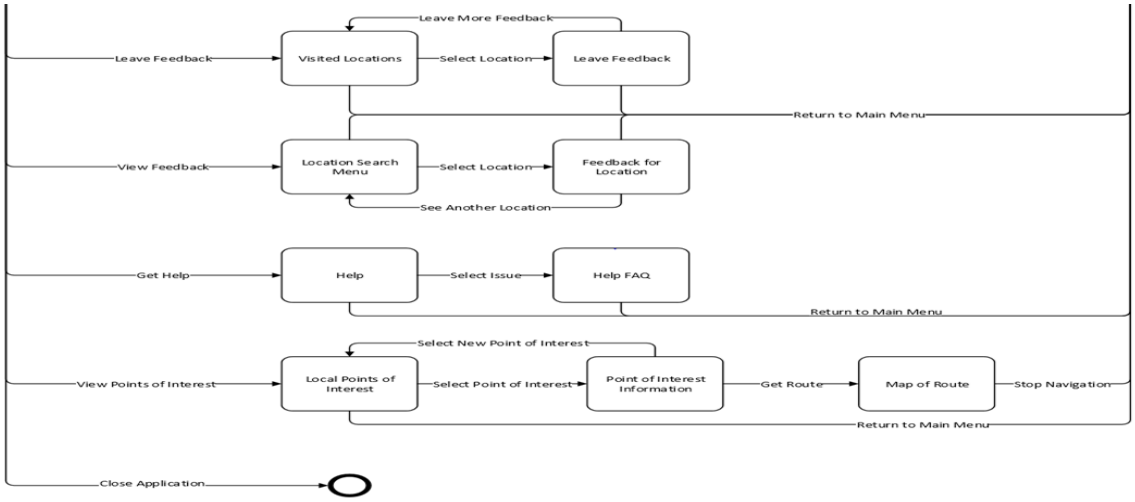
Challenges, Rivals, Leaderboards, and Saved Places:



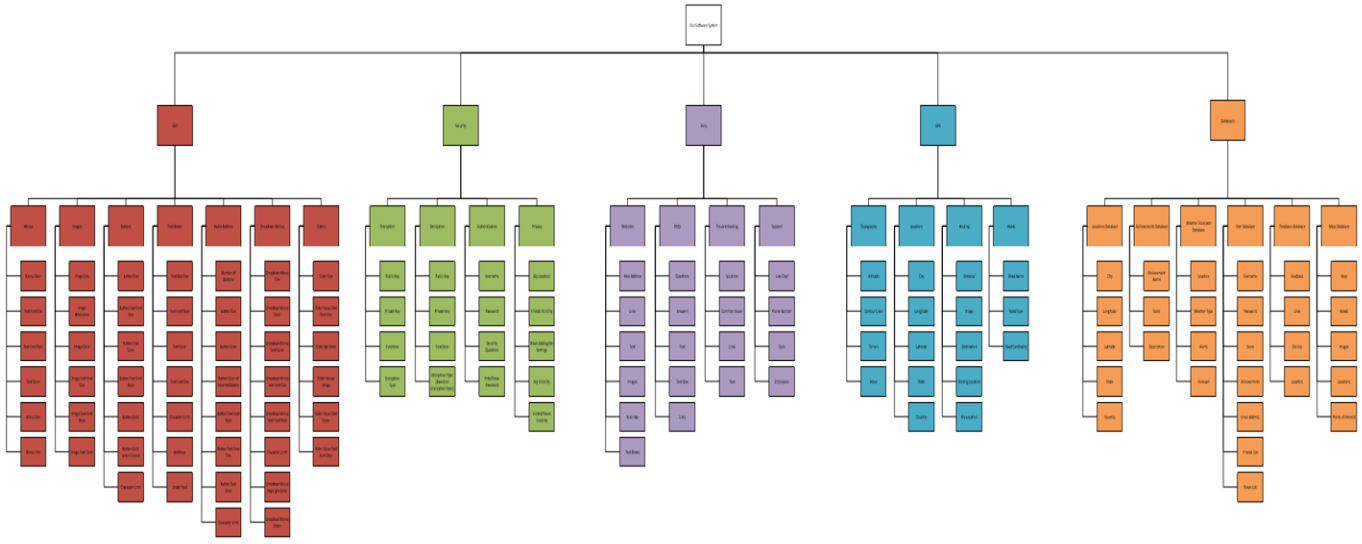
Weather and Account Settings:



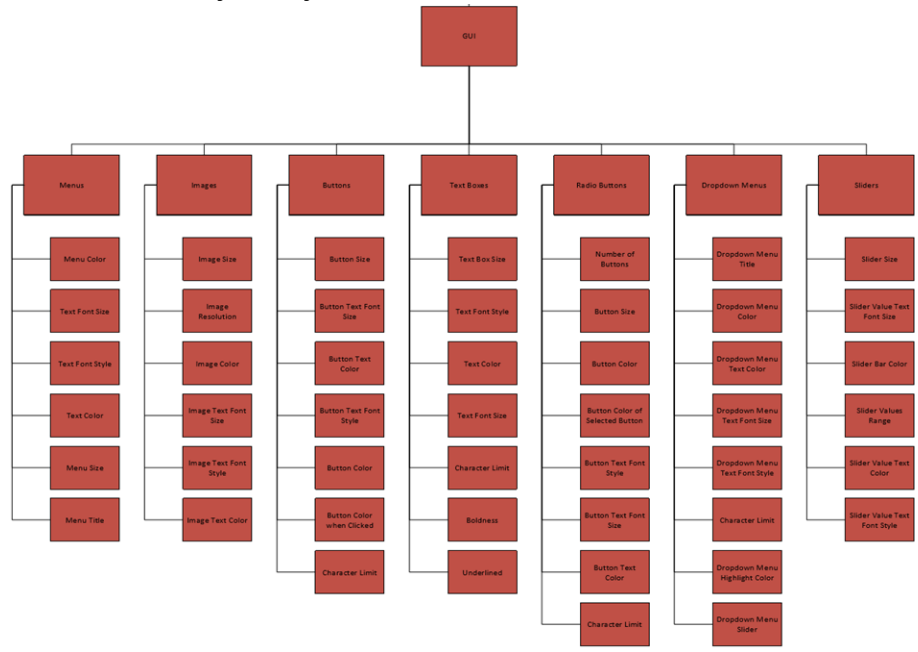
Feedback, Help, Points of Interest, and Close Application:

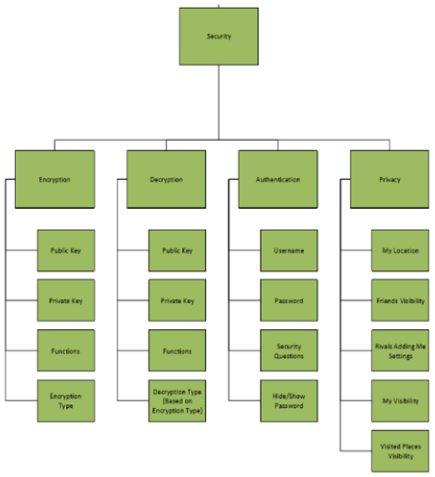


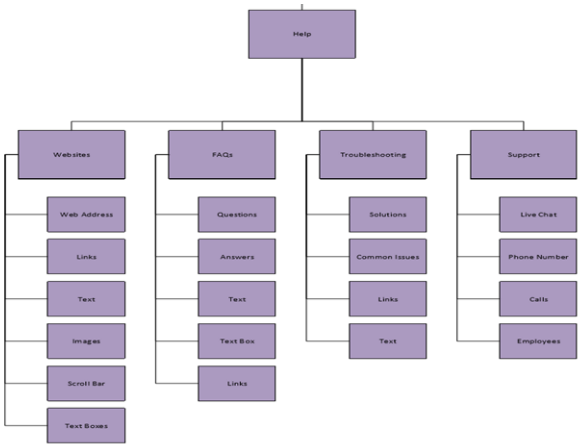
Hierarchical System Architecture Model:

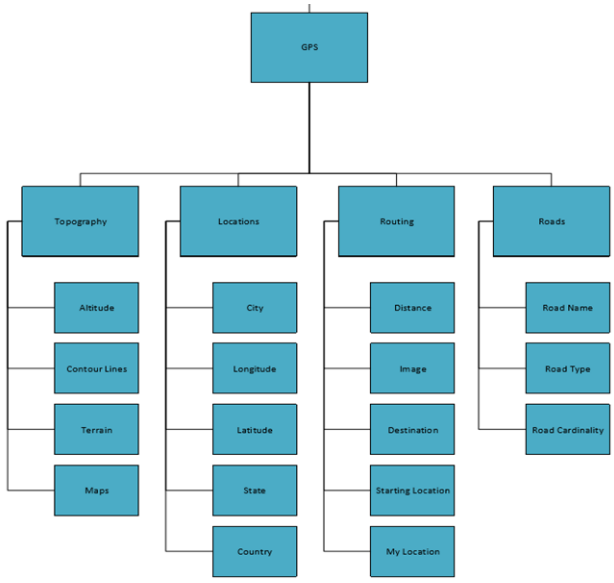


GUI and Security Subsystems:



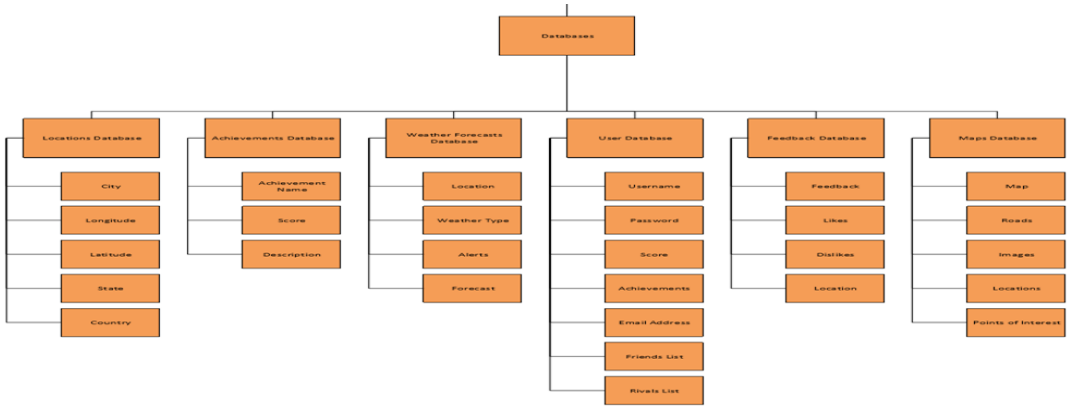




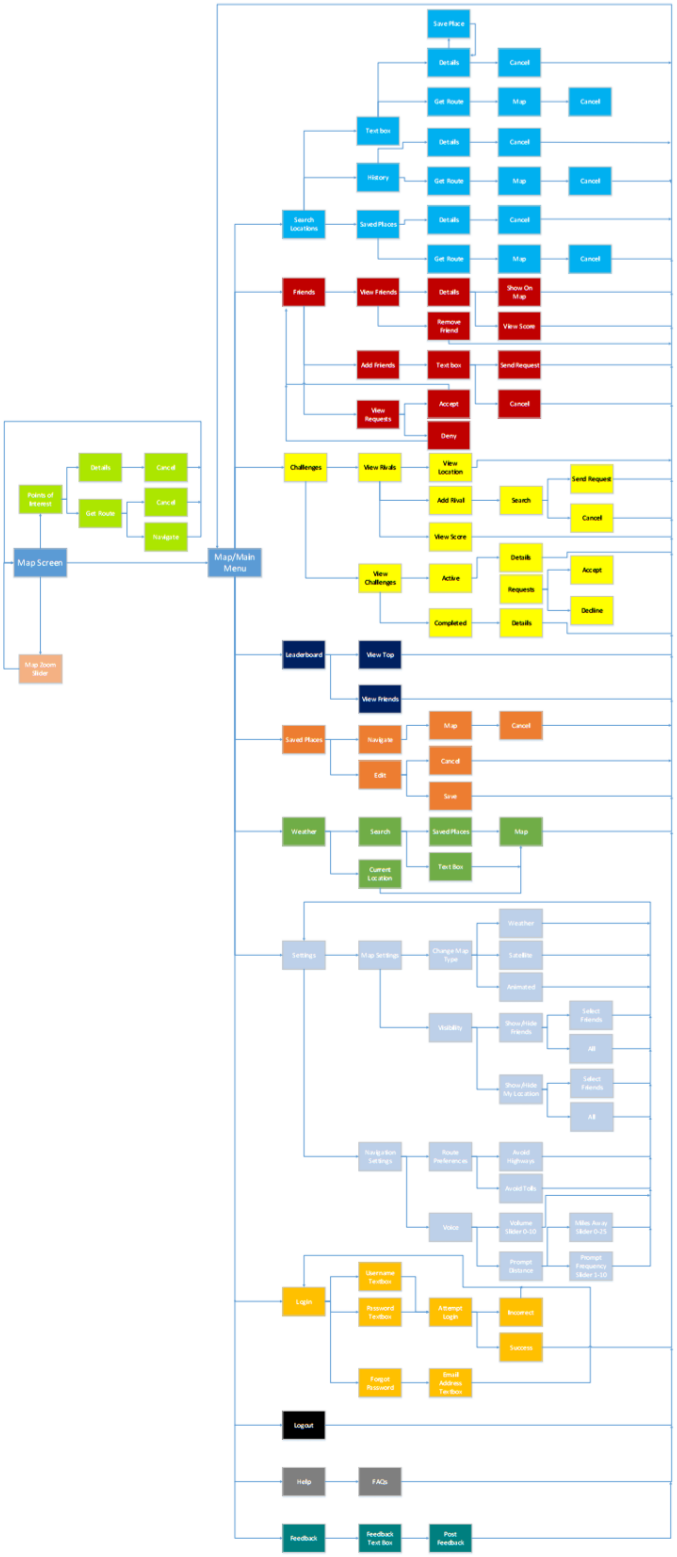


Help and GPS Subsystems:

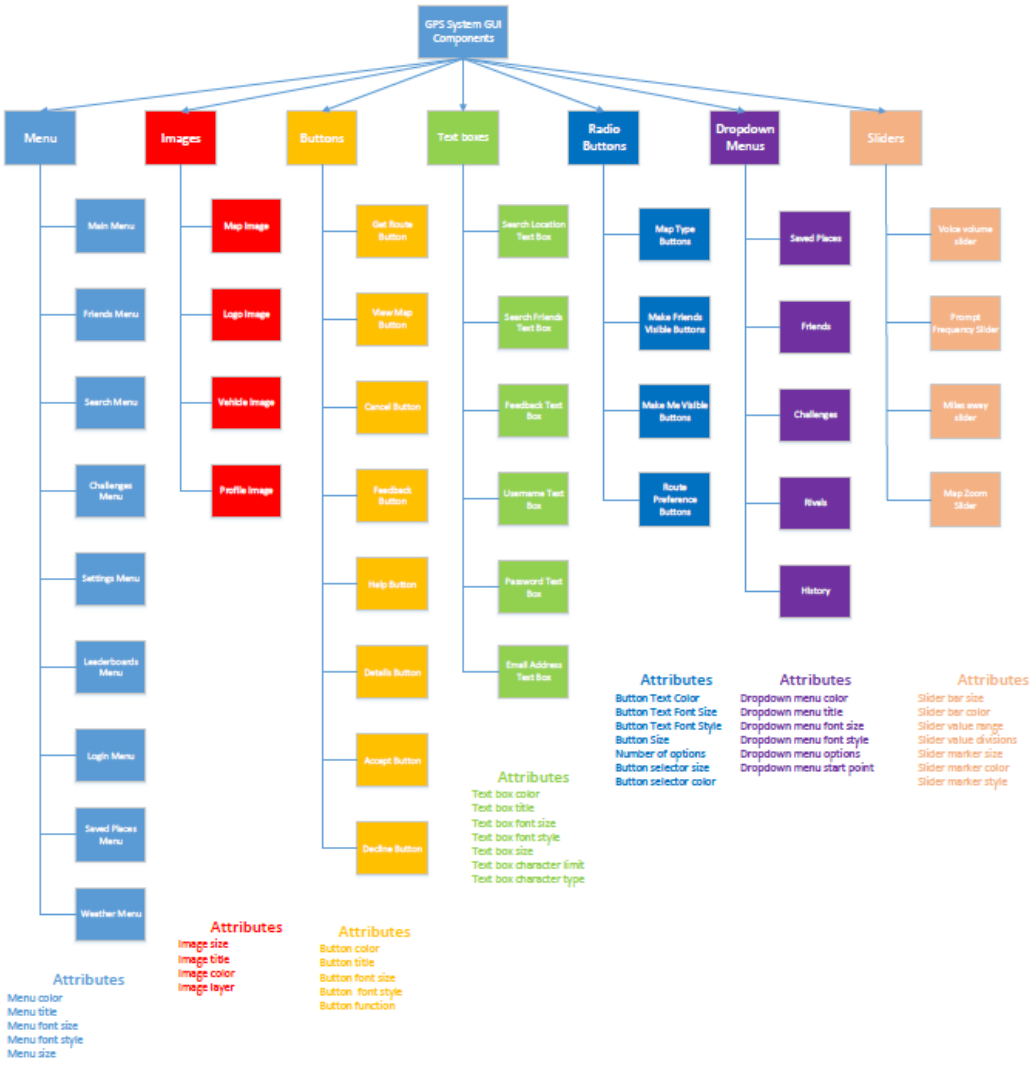
Databases Subsystem:



GUI Functional View:



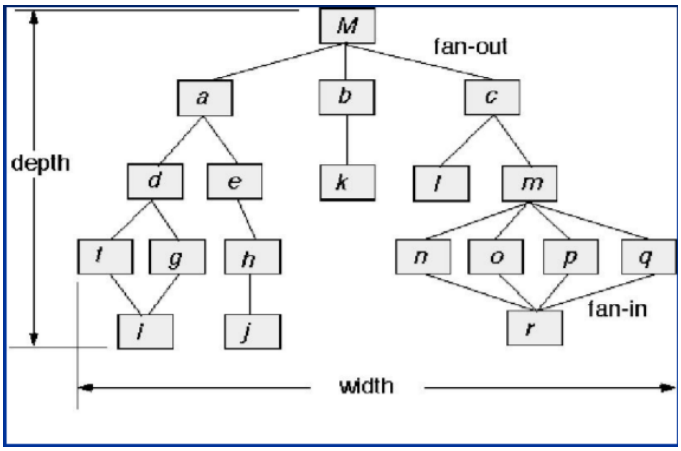
GUI Component View:



Software Architectural Model:

Call and Return Model

- User inputs address to obtain route which is then passed into a subfunction.
- User searches for locations or friends in a data base, and that data is then handled by sub functions.



FP- Based Estimation:

Information Domain Value	Optimal	Most Likely	Pessimistic Estimation	Estimated Count	Weight	FP-Count
Number of inputs	45	60	75	60	4	240
Number of outputs	35	50	80	50	8	400
Number of inquiries	10	15	25	15	3	45
Number of files	8	12	18	12	6	72
Number of external interfaces	4	6	8	6	8	48
Total						805

Number of Function Points: 805

Average productivity: 6.5 FP/month

Burdened Labor Rate: $8000/month

The project will take 124 man months and cost $992,000 to produce.

LOC-Based Estimation:

Function System	Estimated LOC
User Interface	5,000
Route Calculation	35,000
Location Calculation	20,000
Database Management	10,000
Gamification	10,000
Total Estimate	70,000

Lines of code: 70,000

Average productivity: 620 LOC/month

Burdened Labor Rate: $8000/month

The project will take 113 man months and cost $904,000 to produce.

Conclusion:

To summarize, we need a software system with an emphasis on Navigational capabilities that intertwines aspects of today's Social Media applications while also integrating a form of gamification. We should enable the User to get from point A to point B in an enjoyable and user-friendly way that promotes adventurous activity. This project has a budget of $1,000,000 and should be completed between 112-124 man months.