

SquadUp

Dev Doc 1

February 5, 2016

Brandon Hill, Jon Jo, and Eric Grant

Table of Contents

Table of Contents.....	1
Executive Summary.....	2
Project Description.....	3
System Description & Rationale.....	3
System Scope.....	3
System & Development Constraints.....	3
System Services.....	4
Introduction.....	4
Functional Requirements.....	4
Use-Case Diagram Introduction.....	6
Use-Case Diagram.....	7
Use-Case Description.....	8

Executive Summary

Organizing sports activities with larger groups of people can be a hassle and somewhat of a crapshoot. For example, when going to a gym to play basketball there are times when tons of people are playing, and times when there is no one at all. So if there was a way to organize, and view the times people plan on being at the gym, you could avoid playing alone, or playing when it is too crowded.

As a result we have decided to create SquadUp which is an app to make organizing sports activities easier by putting all the information that you need in one place. Our team consists of three computer scientists: Jon Jo (Librarian), Eric Grant (Recorder), and Brandon Hill(Leader). Our client is Brendan Murphy a student at SPU that plays basketball and understands the problem of going to the gym and not being able to (SquadUp) find a pickup game.

Project Description

System Description and Rationale

We are building an app that will organize all the information you need to plan a sports activity with friends. Right now there is no way of knowing for sure who will be at a gym, when they will be at the gym, or how crowded the gym will be. SquadUp will allow users to login and evaluate which gym is the best option for that night. It will also include information such as price so that there are no surprises when they arrive. Our app is for the sports community. There has never been an easy way of organizing a pickup game and we plan on providing that.

System Scope

The SquadUp app will provide users with an easier way of organizing a time and place to meet people, and play basketball. As a result, it will reduce the risk of showing up at a gym with no one to play basketball with. The app will allow users to find locations where they can play basketball, invite friends, add favorite locations, and view when people will be playing. The SquadUp app will also allow organizations to make accounts where they can add and update information about their facility. This includes things such as hours of operation, location, phone number, and more. As of right now, this version of the system will only include basketball. We may expand in a later version to include other sports and activities.

System and Development Constraints

- The app will only be developed for android (no IOS or Windows).
- There is a deadline of May 2015.
- Users will be required to make an account to use the app.

System Services

Introduction

This section will provide the functional requirements, use-case diagram, along with the use-case descriptions for the SquadUp app. The functional requirements specify the services and actions the SquadUp app must be able to perform. The use-case diagram provides a model for the system and documents how the users will interact with the SquadUp app. The use-case description provides a specification for the interactions that occur between the SquadUp app and its actors.

Functional Requirements

Create User Account - The user will be able to create an account and add their name, preferred sports, phone number, and location.

Use Cases: 1 - Create Account

Initial Acceptance Tests: Create an account with no name
Create an account with no sport
Create an account with no phone number
Create an account with no set location
Create an account with the same name and phone number as another account
Create an account with special characters in any field

Create Org Account - Organizations can add their Organization/Gym Name, sports they offer, cost to play, and location.

Use Cases: 1 - Create Account

Initial Acceptance Tests: Create an Org account with no name
Create an Org account with no phone number
Create an Org account with no location
Create an Org account with no hours or pricing
Create an Org account with the same name and phone number as an Org account that already exists

<i>Find Location -</i>	Users can use a map to search for gyms around them or in a specific area.
<i>Use Cases:</i>	2.2 - Find location 2.2.1 - Reserve Location
<i>Initial Acceptance Tests:</i>	Zoom out to a certain distance that the search area is too large Click on a gym location and try to reserve a spot at that gym Click on a gym location and try to check information for that gym Filter gyms on map based on price, hours, busy-ness
<i>Friends List - with</i>	Users can add friends to their network that they play often
<i>Use Cases:</i>	2.3 - Friends List 2.3.1 - View Profile 2.3.2 - Add Friends 2.3.3 - Invite Friends 10: Invite Friends
<i>Initial Acceptance Tests:</i>	Add a username that doesn't exist Add a username that's already in the network Invite a friend Invite friend by sharing a link via Email Invite friend through the app Delete Friend Delete a username that isn't part of your network

Use-Case Diagram Introduction

This section will provide a use-case diagram along with the use-case descriptions for the SquadUp system. The use-case diagram provides a model for the system and documents how the Smart SquadUp system will interact with its environment. The use-case description provides a specification for the interactions that occur between the SmartVend Organics' system and its actors.

Human Actor: A human actor is a user/organization that interacts with the SquadUp system. In the use-case diagram the human actor is represented by the following graphics.



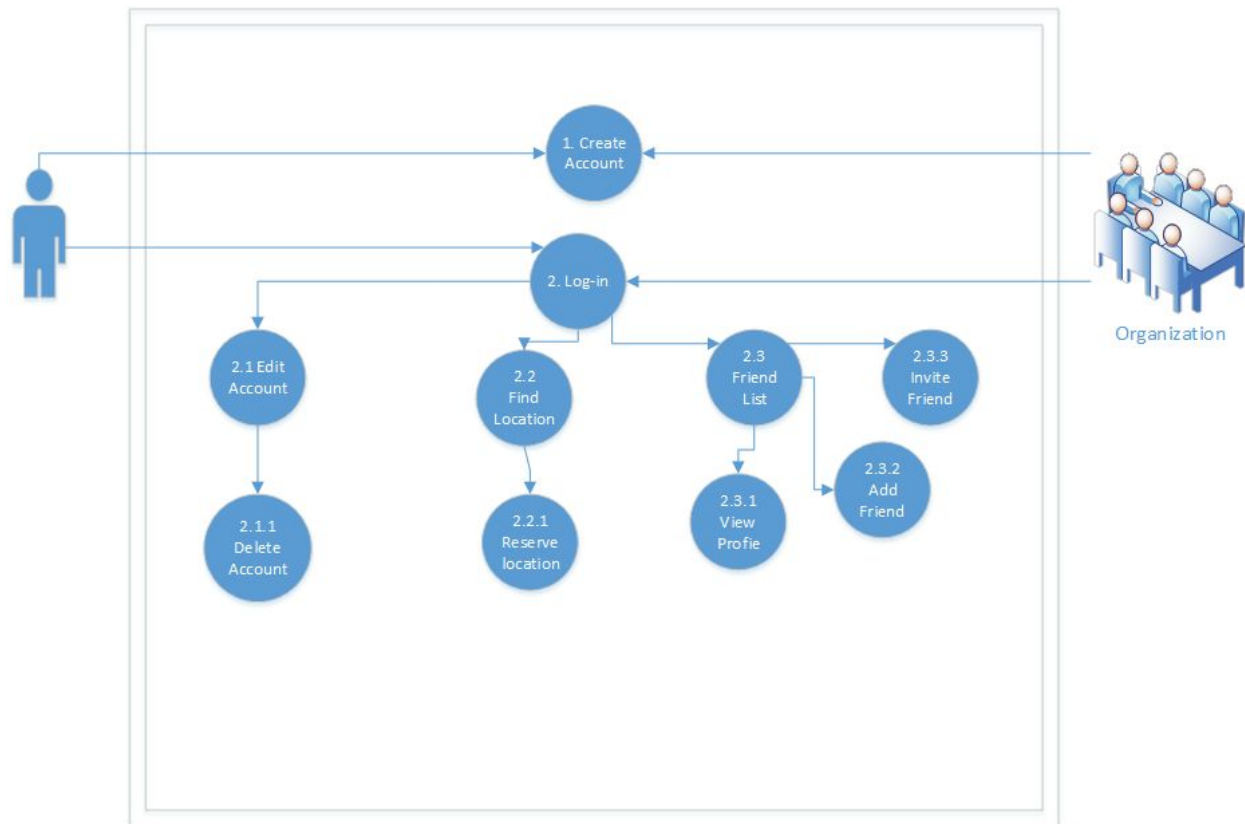
Use Case: A use case characterizes a specific way of using the SquadUp system. In the use-case diagram the use-case is represented by the following graphic.



Association: An association links an actor with a use case with which it interacts. In the use-case diagram the association is represented by the following graphic.



Use-Case Diagram



Use-Case Descriptions

Use-Case: Create Account		ID: 1
Actors: User/Organization	Primary Actor: User/Organization	
Stakeholders and interest: Organization – Wants user to use their location. Users - Wants user as connection for reliable information regarding pick-up games		
Brief description: User or Organization can create account		
Relationships: Association: User/Org Include: Extend: Generalization:		
Trigger: User creates account Type (circle one): External		
Basic Flow of Events: 1.Create Account		
Subflows: 1.a. Adds User Account 1.b. Adds Organization Account		
Extensions / Alternate Flows: 1.a.1. Username already taken 1.b.1 Organization name already taken		

Use-Case: Log in		ID: 2
Actors: Organization/User	Primary Actor: Organization/User	
Stakeholders and interest: User/Organization – Able to access their own account		
Brief description: Organization and user can log into accounts to Edit Account, find location, or access friends list.		
Relationships: Association: Use/Org Include: Extend: 2.1 Edit account, 2.2 Find Location, 2.3 Friends List Generalization:		
Trigger: Organization creates account Type (circle one): External		
Basic Flow of Events: 1. Account login		
Subflows:		
Extensions / Alternate Flows: 1.a1. User name does not exist 1.a2 Password is incorrect		

Use-Case: Edit Account		ID: 2.1
Actors: Organization/User Accounts	Primary Actor: Organization/User accounts	
Stakeholders and interest: User – Can change their login and account information Organization - able to edit account login and account information		
Brief description: User/Organization accounts can be edited by their respective accounts		
Relationships: <div>Association: User/Org</div> <div>Include: 2.1.1 Delete Account</div> <div>Extend:</div> <div>Generalization:</div>		
Trigger: Organization creates account Type (circle one): External		
Basic Flow of Events: 1. Edit Account		
Subflows: 1.a Edit Name 1.b Edit Password 1.c Edit Phone Number 1.d Edit Location 1.e Edit Sport		
Extensions / Alternate Flows: 1.a1. User/Org name already on list 1.a2 User/Org name not found 1.b Password is same as before 1.c1 Location does not exist		

Use-Case: Delete Account		ID: 2.1.1
Actors: Organization/User Accounts	Primary Actor: Organization/User accounts	
Stakeholders and interest: User /Organization - able to delete account if no longer needed or wanted		
Brief description: Organization or user can delete all account information		
Relationships: Association: User/Org Include: Extend: Generalization:		
Trigger: Organization creates account Type (circle one): External		
Basic Flow of Events: 1. Delete Account		
Subflows:		
Extensions / Alternate Flows:		

Use-Case: Find location		ID: 2.2
Actors: Organization/User Accounts	Primary Actor: Organization/User accounts	
Stakeholders and interest: User – able to find and reserve locations Organization - to search their own or other organizations		
Brief description: Organizations and users can find location, access map, and reserve locations if needed.		
Relationships: Association: user/Org Include: 2.2.1 Reserve location, Extend: Generalization:		
Trigger: Organization creates account Type (circle one): External		
Basic Flow of Events: 1. Browse map for nearby gyms 2. Search for specific organization name or city		
Subflows:		
Extensions / Alternate Flows: 1.1. Cannot locate User 2.1 User name not found 2.2 Organization not found		

Use-Case: Reserve Location		ID: 2.2.1
Actors: Organization/User Accounts	Primary Actor: Organization/User accounts	
Stakeholders and interest: User/Organization - able to reserve location while available		
Brief description: Organizations/Users can search and find locations that are available and reserve for a time slot.		
Relationships: Association: User/Org Include: Extend: Generalization:		
Trigger: Organization creates account Type (circle one): External		
Basic Flow of Events: 1. Reserve Location		
Subflows:		
Extensions / Alternate Flows: 1.a1. Location not available		

Use-Case: Friends List		ID: 2.3
Actors: Organization/User Accounts	Primary Actor: Organization/User accounts	
Stakeholders and interest: User – Able to add and invite friends to gyms they are currently present or already reserved. Organization - to add friends and send invites of availability		
Brief description: Organizations and users can view their friends list to add or invite		
Relationships: Association: user/Org Include: 2.3.1 View Profile, 2.3.2 Add Friend, 2.3.3 Invite Friend Extend: Generalization:		
Trigger: Organization creates account Type (circle one): External		
Basic Flow of Events: 1. View Friends 2. View Profile		
Subflows:		
Extensions / Alternate Flows: 2.1 Your friend's profile is on private 2.2 Friend does not exist anymore		

Use-Case: View Profile		ID: 2.3.1
Actors: Organization/User Accounts	Primary Actor: Organization/User accounts	
Stakeholders and interest: User – Able to add and invite friends, see status (gym location), phone number, preferred sport Organization - to add friends and send invites of availability		
Brief description: Organizations and users are able to view other user or organization profiles		
Relationships: Association: User/Org Include: Extend: Generalization:		
Trigger: Organization creates account Type (circle one): External		
Basic Flow of Events: 1. View Account		
Subflows:		
Extensions / Alternate Flows: 1.a1. User name already on list 1.a2 User name not found 1.b1 Organization already on list 1.b2 Organization not found 2.1 User cannot be found 2.1 User cannot be invited (Offline)		

Use-Case: Add Friends		ID: 2.3.2
Actors: Organization/User Accounts	Primary Actor: Organization/User accounts	
Stakeholders and interest: User – Able to add and invite friends Organization - to add friends and send invites of availability		
Brief description: Organizations and users can add other accounts		
Relationships: <div>Association: User/Org</div> <div>Include:</div> <div>Extend:</div> <div>Generalization:</div>		
Trigger: Organization creates account Type (circle one): External		
Basic Flow of Events: 1. Add Account		
Subflows: 1.a Add User 1.b Add Organization		
Extensions / Alternate Flows: 1.a1. User name already on list 1.a2 User name not found 1.b1 Organization already on list 1.b2 Organization not found		

Use-Case: Invite Friend		ID: 2.3.3
Actors: Organization/User Accounts	Primary Actor: Organization/User accounts	
Stakeholders and interest: User – Able to add and invite friends Organization - to add friends and send invites of availability		
Brief description: Organizations and users are able to invite other accounts to current or future reserved locations		
Relationships: Association: User/Org Include: Extend: Generalization:		
Trigger: Organization creates account Type (circle one): External		
Basic Flow of Events: 1. Invite Account		
Subflows: 1.a Invite User 1.b Invite Organization		
Extensions / Alternate Flows: 1.1 User/Org cannot be found 1.2 User/Org cannot be invited (Offline)		