## SEG2105 Final Project Report

**Design and Implementation of Keeper**

Chen, Qufei (6771326)

Langlois, Matthew (7731813)

Yang, Meng (6889352)

Yaraskavitch, Matthew (6301664)

SEG2105 Section B

Due: December 5th, 2015

Presentation Date: Decemebr 4th, 2015

University of Ottawa

Faculty of Engineering

## Introduction

This document serves to formalize the work completed throughout the implementation of the *Keeper* app. The three previous deliverables have been modified based upon marker feedback and the relevant changes listed. The final implementation of these deliverables is listed below. As is common in software engineering, the non-essential requirements have been modified to more closely reflect the final implementation, without changing the intent.

## Contributions and Corrections

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Member** | **Deliverable** | | | |
| **#1** | **#2** | **#3** | **#4** |
| Chen, Qufei | 25% | 25% | 25% | 25% |
| Langlois, Matthew | 25% | 25% | 25% | 25% |
| Yang, Meng | 25% | 25% | 25% | 25% |
| Yaraskavitch, Matthew | 25% | 25% | 25% | 25% |

|  |  |
| --- | --- |
| **Deliverable** | **Changes / Corrections** |
| #1 | * Functional requirements were updated to make specific reference to the “Tournament Maker System” as opposed to the general “system” * The user role “Coach” has been replaced with “Captain” to accurately reflect the final application * Non-functional requirements were updated to more closely reflect the tools used (i.e. Android Studio 1.5 and targeting Android API 23, specify that only the front-end must be written in Java) |
| #2 | * Added knockout, round robin, and combinational classes to class diagram (TODO) * Updated role names and multiplicities (TODO: specify which role names and multiplicities) on class diagram * Removed instance ‘Person’, and changed teamList and team to be generated by the main ‘TournamentMaker’ class |
| #3 | * Replaced the rough mockups from Deliverable 3 with actual snapshots of the UI from the app |

## Software Requirements (Deliverable 1)

### Functional Requirements

1. The Tournament Maker System must allow a player to join only one team.
2. The Tournament Maker System must assure a player has joined a team before participating in a tournament.
3. The Tournament Maker System should inform the players when a goal is scored in one of their matches.
4. The Tournament Maker System must show the team’s rank compared to the other teams participating in the tournament.
5. The Tournament Maker System should allow a player to change their player number.
6. The Tournament Maker System must allow a team to set their avatar.
7. The Tournament Maker System must allow the captain to set the team’s name.
8. The Tournament Maker System must allow the captain to set the team’s logo.
9. The Tournament Maker System must force a team to have a name before joining a tournament.
10. The Tournament Maker System must allow organizers to create new tournaments.
11. The Tournament Maker System must prompt the organizer to choose one format from Round Robin, Knockout, or Round Robin and Knockout combination.
12. The Tournament Maker System must allow the organizer to add, edit, or delete teams from a tournament if it has not started.
13. The Tournament Maker System must allow a player to review previous rounds of play.
14. The Tournament Maker System must display the current tournament standings.
15. The Tournament Maker System must randomize the team listings and order of play.
16. The Tournament Maker System must allow the user to access the instructions.
17. The Tournament Maker System must inform the user that teams cannot be edited once the tournament has started.
18. The Tournament Maker System must allow the user to customize the tournament settings.
19. The Tournament Maker System must store the number of wins and losses for each team.
20. The Tournament Maker System must allow organizers to keep track of a match's score.

### Nonfunctional Requirements

1. The Tournament Maker System must be developed on Android Studio 1.5.
2. The Tournament Maker System must be written using android API level 23.
3. The Tournament Maker System front-end must be written in java.
4. The Tournament Maker System must display changes from the user within 0.5 seconds.
5. The Tournament Maker System must be able to store information for at least 100 teams.
6. The Tournament Maker System back-end must be available 99% of the time.

Use Cases:

(1) Use Case Name: Create a Team

Actors: Coaches

Goals: To create a team

Preconditions: Must be logged in as a coach

Summary: Create a team

Related Use Cases:

Steps:

|  |  |
| --- | --- |
| **Actor Actions** | **System Responses** |
| 1. Click on ‘Add a Team’ | 2. ‘New Team’ page opens |
| 3. Enter team name |  |
| 4. Select logo/display picture |  |
| 5. Press ‘confirm’ | 6. ‘New Team’ page closes, ‘Team Info’ page opens. |

Post-conditions: Team is created.

(2) Use Case Name: Change Team Display Picture

Actors: Coaches

Goals: To change the display picture to a picture from your phone

Preconditions: Team is created, and must be logged in as a coach

Summary: Coach uploads a picture from their phone, and sets it as the display picture/logo of their team

Related Use Cases: ‘Create a Team’

Steps:

|  |  |
| --- | --- |
| **Actor Actions** | **System Responses** |
| 1. Click on ‘My Teams’ | 2. Dialog with list of teams that you belong to opens |
| 3. Click on name of the team you want to edit | 4. Team info page opens |
| 5. Click on the display picture box | 6. Display picture menu appears |
| 7. Choose ‘upload from phone’ | 8. Photo library of phone opens |
| 9. Select picture |  |
| 10. Press ‘confirm’ | 11. Photo library dialog closes, selected picture is displayed on display picture menu |
| 12. Press ‘update display picture’ | 13. display picture menu closes |
|  | 14. redirects back to team info page |

Post-conditions: Profile picture is changed.

(3) Use Case Name: Enter a Tournament

Actors: Coaches

Goals: To enter the team into a tournament

Preconditions:  Team is created, and must be logged in as a coach

Summary: To register the team into a tournament

Related Use Cases: ‘Create a Team’

Steps:

|  |  |
| --- | --- |
| **Actor Actions** | **System Responses** |
| 1. Click on ‘Tournaments’ | 2. Dialog with list of active (open to registration) tournaments opens |
| 3. Click on name of the tournament that you wish to enter | 4. Tournament info page opens |
| 5. Click on ‘Register for the Tournament’ | 6. Registration page opens |
| 7. Choose the team that you wish to enter into the tournament from the drop down list |  |
| 8. Select method of payment |  |
| 9. Check box agreeing to terms and conditions |  |
| 10. Press ‘confirm’ | 11.Dialog with ‘You have been successfully registered’ pops up |
| 12. Press ‘Okay’ | 13. Dialog closes |
|  | 14. Redirects back to main menu |

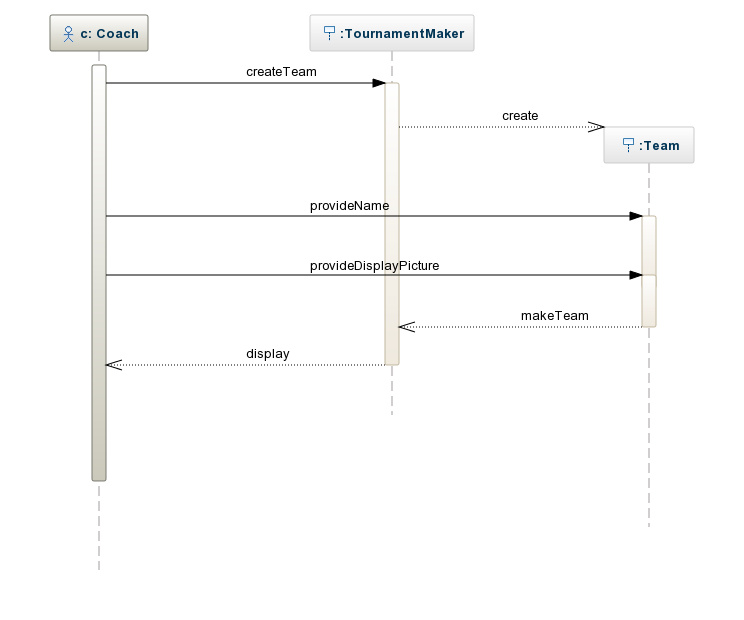
Post-conditions: Team is entered into tournament.

## UML Design (Deliverable 2)

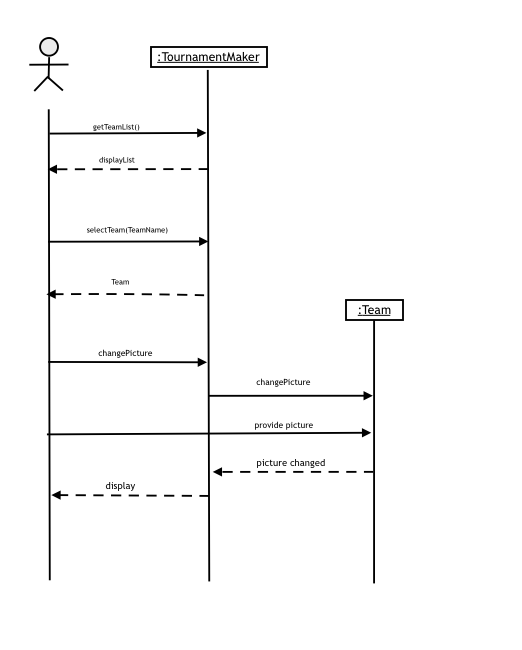
Class Diagram:

Sequence Diagrams

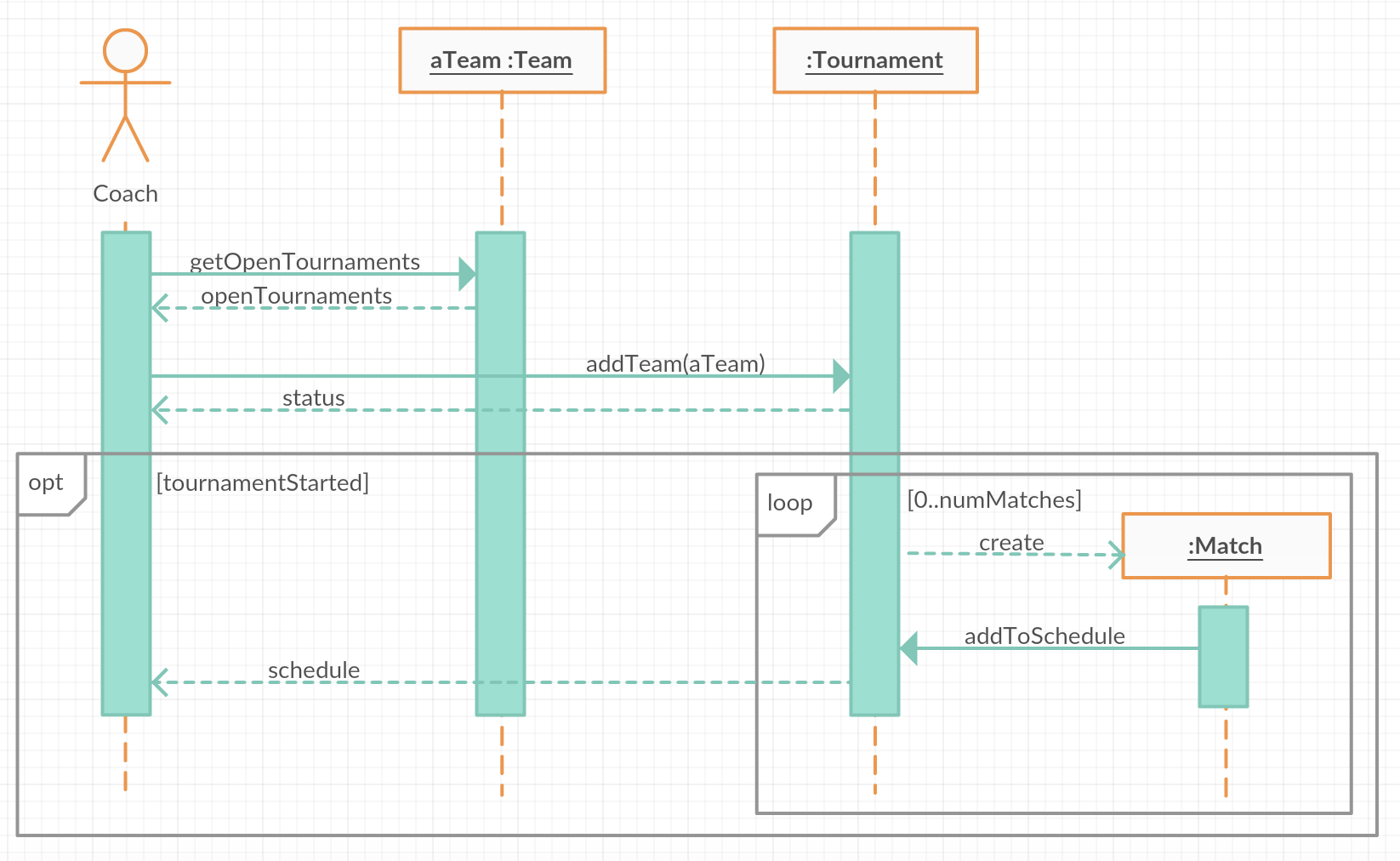
(1) Use case: Create a Team



(2) Use case: Change Team Display Picture



(3) Use case: Enter a Tournament



State Diagram:

Deliverable%202/State_Machine.pdf

## Screenshots of UI (Deliverable 3)