## Project Report

Design and Implementation of Keeper

Chen, Qufei (6771326)

Langlois, Matthew (7731813)

Yang, Meng (6889352)

Yaraskavitch, Matthew (6301664)

SEG2105

December 1, 2015

## 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, Quifei | 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 | TBD |
| #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.

## UML Design (Deliverable 2)

## Screenshots of UI (Deliverable 3)