|  |
| --- |
| **Team Effort**  **Soccer Team**  **Management System**  Milestone 2 |
|  |
| **Luke Brisebois - lub224**  **Simon Fanner - saf725**  **Michael Fulton - mif231**  **Steven Hancock - smh875**  **Adam Mravnik - ajm207**  **Amin Shakev - ams162**  **Patrick Weckworth - paw818**  **Tom Wetzel - thw740**  **Drake Zarowny - djz587** |
|  |

|  |
| --- |
|  |

Table of Contents

1. Introduction 3

1.1 System Description 3

1.2 Current System Status 3

2. Changes From Previous Milestone 3

2.1 Code Improvements 3

2.2 Team Management 4

2.3 Versioning System 4

2.4 Bug Reporting 4

3. Milestone (Revised) 5

3.1 Previous Milestones 5

3.2 Current Milestone 6

3.3 Future Milestones 7

4. Testing 8

4.1 Unit Testing 8

4.2 Build and Smoke Tests 8

4.3 Defect Estimate 8

4.4 Bug Reporting 8

5. Risk Evaluation 8

5.1 Risk Report 8

5.2 Risk Re-evaluation 8

6. Team Personnel 8

6.1 Role Redefining 8

6.2 Lead By Group 9

7. Activity Report 9

7.1 Work Completed 9

7.2 Activity Log 9

8. Design Artifacts 10

8.1 Modified 10

8.2 New 10

9. Conclusion 10

# 1. Introduction

## 1.1 System Description

The project chosen for the class is a Soccer Team Management System, developed by TeamLeader in CMPT 370 in 2011. The software is a web-based system which relies on a database to store and access all types of information. A variety of programming languages are utilized in the project including Java, JavaScript, MySQL, CSS, and HTML. The project was developed in NetBeans IDE and will be done so for this class as well.

## 1.2 Current System Status

The current system is operational and functions as expected, with few bugs. This milestone has seen the development of an extensive testing system using JUnit as well as the improvement of the graphical user interface usablity ability by developing a navigation bar. The team is planning to add to its functionality in upcoming milestones and fix any bugs that are being catalogued and tracked.

There has been a switch version control systems for the project to GitHub and members of Team Effort are currently in the process of familiarizing themselves with the version control tool.

# 2. Changes From Previous Milestone

## 2.1 Code Improvements

An improved user interface is being introduced in Milestone 2. A new navigation bar will allow users to quickly navigate through the applications pages. A user will quickly be able to change from adding a team to adding a player. Quick navigation allows users to perform more actions faster, with fewer clicks. The navigation bar will allow for the application to follow the *three-click rule* that suggests any user of a website/web-application should be able to find the information or perform the action that they are attempting to perform in three clicks or less.

The project’s source code has been transformed into a unified style using TXL. Using TXL all source code, no matter the author and their own styling preferences, can easily be transformed to a single style. The single style will allow for easy readability and prevent the introduction of new bugs due to poor readability when modifying code in addition to aiding the ability to review code and spot bugs that have already been placed in the source code. Team Effort will continue to modify the TXL grammar throughout the project to conform to the group’s agreed upon coding style.

Simple refactoring was used to reorganize source files for images, javascript, and cascading style sheets. The reorganization provided better directory structure and improved readability. Which with a lot of contributors to one single product can go a long way in reducing future bugs.

## 2.2 Team Management

Team management has changed both in terms of leadership and in terms of philosophy. Group members will take turns being the group leader, however, the role of the group leader has changed. The group leader will perform the tasks that are required prior to submitting the milestone such as finalizing the milestone document, running TXL on the source code, performing a MySQL dump on the database, and the actual submission of the milestone. The change to the requirements of the group leader role has occurred because of a change in philosophy. Team Effort believes consensus among all group members is extremely important and thus, instead of being lead by a dictatorship, the group will lead by building consensus. The details of the consensus building will be discussed further later in this document.

## 2.3 Versioning System

Team Effort originally planned on using SVN as its project’s versioning control system. However, over several weeks, it became apparent that SVN would not be usable by the group. The SVN repository set up by Computer Science IT was not accessible by group members through off campus computers as a result of SSL Certificate errors. After filling bug reports with Computer Science and several attempts to resolve the problem, Team Effort decided that the time spent trouble shooting could be better spent else where in the project, so we migrated from SVN version control, to GitHub. GitHub presented its own issues as no member of the group has used it before and documentation is not as extensive as SVN, however so far, there have been no issues.

## 2.4 Bug Reporting

The team’s current bug tracking system was simply recording the bug. A new system has been developed that provides a checklist of important information that must be filled out for each and every bug. Most bug reporting systems are deployed on a Web Server to provide easy access to users, however Team Effort’s bugs will be reported through their Wiki page for simplicity and to avoid overhead.

**Title**: *Example Bug*

**Priority**: *Must fix |* *Will fix opportunistically |* *Desirable, but improbable* | *Extremely improbable*

**Severity**: *Data loss or security issue | Major functionality doesn't work as specified |* *Minor functionality doesn't work as specified*

**Reproduction formula**:

*To reproduce this example bug…*

**Assigned to person**: *Someone*

**Area of the project**: *What page? What Servlet? Etc*..

**Opened by person**: *Who found this*

**Status**: *Active |* *Fixed* *| Resolved | Closed*

**Resolved as**: *Fixed | Postponed |* *Duplicate | Won't be fixed*

**Type**: *(Regular/Regression)*

**Triage**: *whether triaged, if so accepted/rejected*

# 3. Milestone (Revised)

There will be five major Milestone deliverables and five presentations over the course of the term. Each Milestone will include some new functionality as well as its related testing. Below is a Mini-Milestone list of the proposed objectives. Future milestones objectives have been altered to better reflect the need for management and tool implementation as opposed to design implementation. Previous milestones objectives have not been altered, however conclusions have been added, all other milestones are being refined in this document and in future milestones, removed object are in red, added objectives are in green.

## 3.1 Previous Milestones

**Milestone 1**

*Objectives*

* Resolve technical issues with database and versioning system
* Begin detecting existing bugs, defects, and limitations of program
* Develop plan for subsequent milestones

*Conclusions*

* Had previous database restored and migrated to Team Effort's database.
  + Made backup copies, and will continue making backup copies to ensure problem doesn't arise again.
* An SVN repository was set up, but accessibility was difficult. For Milestone 1 we used drop box until our SVN accessibility issues were fixed.
  + SVN as a version control system has been dropped in the current Milestone (Milestone 2) in favor of GitHub.
* Bug detections and determining application limitations is ongoing.
* Revising plans for upcoming milestones is ongoing.

*Time Sheet*

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Time Spent | I worked on… |
| Luke Brisebois | Jan 24 | 1 Hr | Reviewing Soccer Project  -running the system  -looking at code  -familiarizing myself |
| Feb 2 | 2 Hr | Adding to report |
| Simon Fanner | Jan 31 | 1 Hr | Report/Presentation Outline |
| Feb 1 | 2 Hr | Report documentation/Tech issues |
| Feb 2 | 2 Hr | Database dump/Report completion  - handed in Milestone 1 to moodle  - may need/have time to revise it |
| Michael Fulton |  |  |  |
| Steven Hancock | Feb 1 | 1.5 Hr | TXL Pretty Printing |
| Feb 1 | 1.5 Hr | Risk Analysis |
| Feb 3/4 | 0.5 Hr | Just a little bit more on TXL, and a small write-up |
| Adam Mravnik | Feb 1/2 | 2 Hr | Tried to resolved ssl and access issues with svn  by communicating with CS Trac and modifying my own SSL settings. |
|  | Feb 2 | 0.75 Hr | - Restructured the project files in Dropbox.  - Removed all old svn references from when it was a 370 project.  - Renamed the project and various files to make their purpose updated  and more clear.  - Imported the project into the SVN repo using Tortoise SVN  - Current SVN Status: Works in windows, and on tuxworld (possibly all linux distros?),  does not seem to work in OSX |
| Amin Shaker |  |  |  |
| Patrick Weckworth |  |  |  |
| Tom Wetzel | Jan 18 | .25 Hr | Setting up wiki pages |
| Feb 1 | 2 Hr | Working on presentation and milestone goals |
| Feb 2 | 0.5 Hr | Helping connect to SVN and the database, presentation meeting |
| Drake Zarowny |  |  |  |

## 3.2 Current Milestone

**Milestone 2**

*Objectives*

* Ability for players and managers to edit their own user profile
* Ability for users to edit their information and password
* Tool Bar/Menu for navigation
* Support for more than one league/division
* Implementation of unit testing using Junit
* Develop a more extensive bug tracking tool
* Use TXL to format code Java code into single style
* Use of refactoring in Netbeans in order to better structure code
* Restructure and refine database (in code and in database)
  + Revise password authentication (more secure, terminate sessions)

## 3.3 Future Milestones

**Milestone 3**

*Objectives*

* Revise password authentication (more secure, terminate sessions)
* Change date selector format and input to something more user friendly
* Implement NiCad to determine software clone's in the project
  + Categorize some clones and the possible reason for the existence
  + Remove some clones from the system
* Implement logging feature
* Implementation of additional tools like Checkstyle, TeamCity, jMock, Apache Maven, etc

**Milestone 4**

*Objectives*

* Make system more user-friendly
* Auto scheduling system
* More efficient stat tracking
* Use VisCad to help visualize, manage, and order clones detected by NiCad
* Implementation of additional tools like Checkstyle, TeamCity, jMock, Apache Maven, etc

**Milestone 5**

*Objectives*

* Different view styles on profiles or stats (sort by teams etc)
* Player email notification
* More robust messaging system
* Use VisCad to further refine additional clones, identify critical clones and determine why they should or should not exist.
* Implementation of additional tools like Checkstyle, TeamCity, jMock, Apache Maven, etc

# 4. Testing

## 4.1 Unit Testing

## 4.2 Build and Smoke Tests

## 4.3 Defect Estimate

## 4.4 Bug Reporting

# 5. Risk Evaluation

Below is a list of requirements from the previous milestone. Relevant comments on risks have been placed in blue, while amendments or added risks are in red.

## 5.1 Risk Report

*Software Requirement Risks*

* Change of requirements
  + The requirements of the software have drastically changed. Team Effort has removed many of the planned additions that we laid out in the first milestone and shifted focus towards implementing new tools to help manage the software.
* Poor definition of requirements
  + The requirements from the previous milestone were some-what vague and had to be further revised for this milestone such as “What is a navigation bar? What should it include?” and “What is restructuring the database?”
* Impossible requirements
  + All requirements were possible

*Software Risks*

* Project & Milestone completion dates being unrealistic
  + Realistic given an isolated class, however with many assignments due and Mid-terms around the same time as Milestone 2 is due makes allocating time to this assignment difficult.
* Hardware (server issues)
  + Not a problem
* Lack of Testing
  + Not a problem
* Personal differences in design/coding techniques
  + Since there is less focus on design and more focus on management this risk is easily avoidable, especially with the help from TXL to format the code in one singular way.
* Lack of knowledge could make some features unobtainable
  + Not a problem
* Human Errors
  + None have occurred

*Software Scheduling Risks*

* Over-estimate time requirements
  + Not a problem
* Under-estimate time requirements
  + The group has severely under-estimated the time it requires to coordinate activities and to split up the work. Weekly meetings have not been enough, especially when its not always possible for everyone to make the meetings.
* Not managing time affectively
  + Because there is an under-estimation of time requirements the group needs to work better at maximizing the time that they do have together. Things like being better prepared for every meeting and participating on the project’s Wiki will help.
* Requirements changing and not being able to adequately allocate time
  + Not a problem
* Lack of skill could require additional learning to implement goals
  + Not a problem
* Tool failure, like SVN, or difficulties with NetBeans or Java Server
  + This has been the team’s single biggest hurdle. SVN has given Team Effort headache’s and the issue took weeks to resolve, and the resolution is still a little ‘iffy’. In the migration to GitHub there have also been difficulties as learning on the fly how to use a new version control system is not ideal.
* Lack of knowledge of tools
  + The project management tools that Team Effort would like to are all to new to the group. So time will have to be allocated for individual members (and the group as a whole) to learn the new tools in order to implement them into the project.

*Software Quality Risks*

* Improper or lack of design documentation
  + Not a problem
* Unrealistic scheduling leading to lack of testing and deploying bug filled application
  + Not a problem
* Lack of knowledge leading to unforeseen bugs, errors, or unexpected results
  + The group has limited knowledge of new tools. The new tools that are being introduced to the project bring the possibility of introducing new errors, bugs, and unexpected results.
* Application’s user interface not easy to use
  + This is a minor issue, however a navigation bar, would help to make the interface easier to use. For this reason, a navigation bar will be implemented in this milestone.

*Team Risks*

* Lack of communication
  + Communication is becoming a problem. Participation in the Wiki is not as active as it could be, classes and meetings are not being attended by all, and outside of class, group meetings, and the Wiki sees little to no communication.
* Scheduling conflicts
  + It has been very difficult for Team Effort to arrange any meeting times outside of the already scheduled weekly meeting.
* Lack of responsibility (ownership)
  + Those participating are taking ownership.

*Software Business Risks*

* No one wants the application
  + This is not applicable as we are not trying to sell/distribute the application.
* Budget failure (time or financial)
  + It remains unclear if the budgeted time we have given will create a project failure.
* Distribution failure
  + Distribution will most likely not occur in the form of Developer to User.

## 5.2 Risk Re-evaluation

In light of our possible risks, the team will take a number of actions to ensure maximum risk avoidance. The requirements will be outlined as complete as possible in the first Milestone, however, Team Effort will remain flexible in case new requirements, or requirement definitions are changed. Any requirements that are deemed to be too expensive or impossible to meet will be abandoned immediately and no more time will be allocated to those requirements. Team effort will always conservatively budget time to ensure time-cost over-runs are minimal. The team will also ensure sufficient time is allocated for testing. Through code reviews and TXL ‘pretty printing’ the software’s code, although written by multiple programmers, will all be uniform.

# 6. Team Personnel

For the second milestone the team’s roles have been redefined in addition to the leader, mostly a figure head, also changing. The leader in this milestone will be Steven Hancock who will be the main contact person for the group as well as in charge of handing in the milestone.

## 6.1 Role Redefining

Roles have been redefined so that every group member will have the opportunity to enrol in every role for at least one milestone. However, there has been an adjusted of the number of coders and those performing the testing and reviewing. Because the focus has changed from development to testing we have added two Testers/Reviewers and removed two coders. The redeployment of roles for this milestone will allow us to implement extensive JUnit testing. Team Effort’s roles are as follows.

|  |  |
| --- | --- |
| Role | Name |
| Documentation/Presentation (& Leader) | Steven |
| Documentation/Presentation |  |
| Documentation/Presentation |  |
| Coding | Tom |
| Coding | Adam |  |
| Testing/Peer Review | Amin |  |
| Testing/Peer Review | Simon |  |
| Testing/Peer Review | Patrick |  |
| Testing/Peer Review |  |  |

## 6.2 Lead By Group

During Milestone one Team Effort was able to reach important decisions by forming a group consensus. As of such, Team Effort believes that having a static group leader does not conform to the reality of the group’s structure. Consensus building removes potentially unpopular, non-scrutinized, dictated decisions and ensures every important decision is thoroughly discussed and agreed upon by the entire team.

Important decisions will be discussed in group meetings, *Consensus Through Discussion*. As an example Team Effort had difficulties with the SVN server that was hosting the Soccer System project. The group was facing the prospect of having a project that had no version control system. During a group meeting, members of Team Effort discussed the pro’s and con’s of changing version control systems and came to a consensus during the discussion that GitHub should be used instead of SVN. As a result of consensus through discussion the entire group was able to come to a single decision through input from all group members.

*Consensus Through Editing* will govern consensus building for minor decisions. For decisions that have to be made quickly and have little affect on the rest of the project there is not time to discuss during group meetings. For instance, take the decision to include this paragraph in this document. The author had to develop it on the fly, with little input from other group members. However, if this paragraph makes it to the final revision of this milestone it has been agreed to by consensus through editing. Any decision that is not disputed or reverted by a group member other than the author can assumed to be agreed upon by consensus. This is because of our extensive review process, every decision will be reviewed, whether that is code or documentation, by at least one peer, if that peer does not edit the decision that was made in that code or documentation it is assumed that that peer agrees with the decision made by the author. This principle is at the foundation of Wiki documentation.

# 7. Activity Report

## 7.1 Work Completed

## 7.2 Activity Log

Place GitHub Log here.

# 8. Design Artifacts

## 8.1 Modified

## 8.2 New

# 9. Conclusion