**Software Design Document**

**For Target Finder**

Devin Taniguchi

Rodolfo Diaz

Zijun Zhao

Contents

[**1.** **Introduction** 3](#_Toc451604165)

[**2.** **System Overview** 3](#_Toc451604166)

[**3.** **System Architecture** 3](#_Toc451604168)

[**3.1** **Assumptions** 3](#_Toc451604169)

[**3.2** **Constrains** 3](#_Toc451604170)

[**3.3** **System environment** 3](#_Toc451604171)

[**3.4** **Architecture** 3](#_Toc451604172)

[**4.** **Data Design** 3](#_Toc451604173)

**4.1 Data Structure**...................................................................................................................5

**4.2 Database**………………………………………………………………………………….5

[**5.** **Component Design** 3](#_Toc451604174)

**5.1 User Component**…………………………………………………………………………7

**5.2 Maze Code Component**…………………………………………………………………7

**5.3 A.I. Component**…………………………………………………………………………7

**5.4 Directional Code Component**………………………………………………………….7

**5.5 Micromouse Code Component**…………………………………………………………7

[**6.** **Human Interface Design** 3](#_Toc451604175)

[**7.** **Requirements** 3](#_Toc451604176)

1. **Introduction**
2. **System Overview**

The Target Finder system provides a simulation for the path-finding of a mouse reaching the destination in the shortest steps. The system has four layers, including user interface, user interface management (Authentication and Authorization), application functionality, system support (database). The user interface is used by clients to interact with the data system. User authentication is used to verify the users. The application functionality provides the functionality for the system. The database system manages a lot of different maze information.

1. **System Architecture**
   1. **Assumptions**

The user of the Target Finder system should have a knowledge of the basic operations of a computer and the system.

* 1. **Constrains**

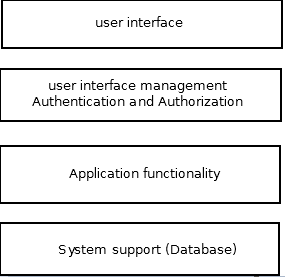
The Target Finder system is designed to create only for the robot company. The system is designed and implemented using Java.

* 1. **System environment**

The Target finder system is designed to work perfectly in any operating system such as Windows, Linux, and OSX. It is also designed to work through any computer and laptop.

* 1. **Architecture**

The layered architecture demonstrates the architecture of Target Finder system.



The Target Finder system has four systems, including user interface, user interface management (Authentication and Authorization), application functionality, system support (database).

1. The top layer is responsible for the user interface. User could interact with the interface. The interface will give the user a guideline to guide user how to edit the inputs and get expected output. The user interface will implement using Java Gui.
2. The second layer is responsible for user to log in to the system. When customers want to access the system, authentication is utilized to identify exactly who is accessing the program, and authorization is utilized to identify if the user has permission to use this resource.
3. The third layer provides the application functionality of the system. It interact with user’s input and generate the expected output.
4. The last layer is responsible for database system Support. The data system is designed to deliver maze information to the user interface when user select a maze. User interface will generate the maze according to the maze information. Also it provides database management system.
5. **Data Design**
   1. **Data Structure**

In Target Finder, the data structure implemented within Java were Map and HashMap. These two data structures, alongside various Java libraries listed in Section 7, were used to implement all six different components (Direction, Maze, Micromouse, AI, User, Map input/output files) within the system. The implementation of Map and Hash maps allows Target Finder to store various forms of data from which the system generates.

* 1. **Database**

The database used in Target Finder were the various libraries accessible within the Java Developer Kit 1.8 package. These libraries were primarily used in order to generate a user interface in which the user can use to maneuver their mouse within a maze. The libraries used in Target Finder are listed in the Requirement Section 7a.

1. **Component Design**

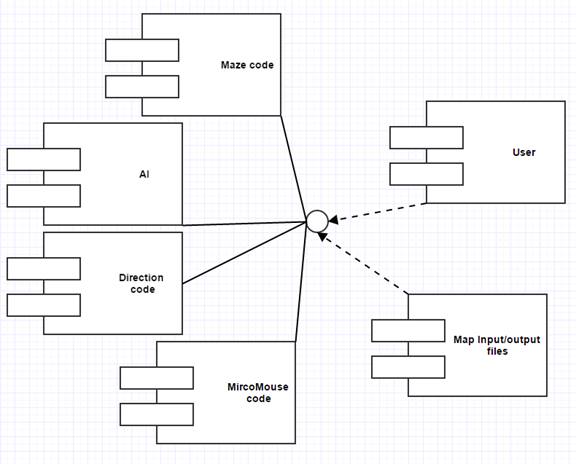


Figure 5.1

Target finder’s component design diagram (figure 5.1) shows the relationship between: the user, MicroMouse code, Maze code, Artificial Intelligence component, Direction Code, the Map input/output files.

**5.1 User Component**

The User component has access to the systems interface. This interface is connected into the other five sub-components in order to input the users cursor direction while moving across mazes.

**5.2 Maze Code Component**

The maze code component takes in user input to generate a maze that the user is going to use. This inputted information dictates the scale of the maze alongside the speed in which the user chooses to maneuver within the maze.

**5.3 A.I Component**

The Artificial Intelligence (AI) component accesses the maze code, map code, and micromouse code to move the cursor throughout the map in the shortest steps possible. These steps are then sent to the interface to allow the user to see an the artificial process of human learning.

**5.4 Direction Code Component**

The Direction code component takes in the user’s cursor direction to dictate which way the mouse within the maze should move. This directional component will rely on the information given within the micromouse component so that the mouse may move within the maze.

**5.5 Micromouse Code Component**

The micromouse component will generate a mouse object within the maze so that the user can maneuver throughout the maze of their choosing.

1. **Human Interface Design**
2. **Requirements**