

SE 3XA3: Software Requirements Specification Mastermind

Team 204, Trident Inc.
Justin Rosner, rosnej1
Justin Prez, prezj
Harshil Modi, modih1

April 4, 2020

Contents

1	Project Drivers	1
1.1	The Purpose of the Project	1
1.2	The Stakeholders	1
1.2.1	The Client	1
1.2.2	The Customers	1
1.2.3	Other Stakeholders	1
1.3	Mandated Constraints	2
1.3.1	Solution Constraints	2
1.3.2	Implementation Environment of the Current System	2
1.3.3	Partner or Collaborative Application	2
1.3.4	Off-the-Shelf Software	2
1.3.5	Anticipated workplace Environment	3
1.3.6	Schedule constraints	3
1.3.7	Budget constraint	3
1.4	Naming Conventions and Terminology	4
1.5	Relevant Facts and Assumptions	4
1.5.1	Relevant Facts	4
1.5.2	Relevant Assumptions	5
2	Functional Requirements	6
2.1	The Scope of the Work and the Product	6
2.1.1	The Context of the Work	6
2.1.2	Work Partitioning	7
2.1.3	Individual Product Use Cases	7
2.2	Functional Requirements	8
3	Non-Functional Requirements	9
3.1	Look and Feel Requirements	9
3.1.1	Appearance	9
3.1.2	Style	9
3.2	Usability and Humanity Requirements	10
3.2.1	Ease of Use Requirements	10
3.2.2	Personalization and Internationalization Requirements	10
3.2.3	Learning Requirements	10
3.2.4	Accessibility Requirements	11
3.3	Performance Requirements	11

3.3.1	Speed and Latency Requirements	11
3.3.2	Safety-Critical Requirements	11
3.3.3	Precision or Accuracy Requirements	12
3.3.4	Reliability and Availability Requirements	12
3.3.5	Robustness or Fault-Tolerance Requirements	12
3.3.6	Capacity Requirements	12
3.3.7	Scalability or Extensibility Requirements	13
3.3.8	Longevity Requirements	13
3.4	Operational and Environmental Requirements	13
3.4.1	Expected Physical Requirements	13
3.4.2	Requirements for Interfacing with Adjacent Systems	13
3.4.3	Productization Requirements	14
3.4.4	Release Requirements	14
3.5	Maintainability and Support Requirements	14
3.5.1	Maintenance Requirements	14
3.5.2	Support Requirements	14
3.6	Security Requirements	15
3.6.1	Access Requirements	15
3.6.2	Integrity Requirements	15
3.6.3	Privacy Requirements	15
3.6.4	Audit Requirements	15
3.6.5	Immunity Requirements	15
3.7	Cultural and Political Requirements	16
3.8	Legal Requirements	16
3.9	Health and Safety Requirements	16
4	Project Issues	16
4.1	Open Issues	16
4.2	Off-the-Shelf Solutions	16
4.3	New Problems	17
4.3.1	Effects on the Current Environment	17
4.3.2	Effects on the Installed Systems	17
4.3.3	Potential User Problems	17
4.3.4	Limitations in the Anticipated Implementation Environment That May Inhibit the New Product	17
4.3.5	Follow-Up Problems	17
4.4	Tasks	18
4.4.1	Project Planning/Development Process	18

4.5	Migration to the New Product	18
4.5.1	Requirements for Migration to the New Product	18
4.5.2	Data That Has to be Modified or Translated For the New System	18
4.6	Risks	18
4.6.1	Excessive Schedule Pressure	18
4.6.2	Management Malpractice	18
4.6.3	Low Quality	18
4.7	Costs	19
4.8	User Documentation and Training	19
4.8.1	User Documentation Requirements	19
4.8.2	Training	19
4.9	Waiting Room	19
4.10	Ideas for Solutions	20
5	Appendix	21
5.1	Symbolic Parameters	21

List of Tables

1	Revision History	iii
2	Work Partitioning	7

List of Figures

1	Context of the Work.	6
2	Use Case Diagram.	8

Table 1: Revision History

Date	Developer(s)	Change
2020-01-22	Justin Rosner, Justin Prez, Harshil Modi	Initial write-up of SRS document
2020-04-04	Justin Rosner, Justin Prez, Harshil Modi	Revision 1 of SRS

1 Project Drivers

1.1 The Purpose of the Project

The purpose of this project is to recreate the famous 1970s game Mastermind with a user-friendly graphic user interface (GUI), and additional levels of difficulty to captivate the user. After the initial hype of the game in the 1970s, the number of users that play the game has decreased drastically. With the new and improved design, our team aims to make Mastermind a game that is played and enjoyed by everyone once again.

1.2 The Stakeholders

1.2.1 The Client

The clients of our project are ~~academia/professional psychologists and the final commissioners that will be making profit of the deployment of the app. Academia and professional psychologists will use the new and improved game as part of the IQ test instead of the older version. Additionally, final commissioners include researchers interested in the link between mastermind and IQ.~~ the TA's of SFWRENG 3XA3, and Professor Bokhari as they are ~~commissionin~~ the project

1.2.2 The Customers

In this case the customers ~~are the~~ include the clients plus the end users of the product. The end user includes individuals of all ages over the MIN_AGE and can obtain a computer/tablet. Mastermind will be free to obtain by anyone through the Google play store or Apple app store.

1.2.3 Other Stakeholders

Additional ~~stakeholder~~ stakeholders of the product include McMaster university, the development team consisting of Harshil Modi, Justin Prez, Justin Rosner, as well as ~~academia/professional psychologists and the final com-missoners that will be making profit off of the development of the app. Academia and professional psychologists will use the new and improved game as part of an IQ test instead of the older version of the game.~~ Furthermore,

final commissioners include researchers interested in the link between Mastermind and IQ.

1.3 Mandated Constraints

1.3.1 Solution Constraints

- **Description:** The product shall be operational on both iOS and Android devices.
Rationale: iOS and Android are the main operating systems used by our customers and clients.
- **Description:** The app should be downloadable through the Apple app store and Google play store.
Rationale: Customers and clients will have Android or iOS devices.
- **Fit Criterion:** The product should be compatible with both mobile and tablet devices of varying size.
Rationale: Clients will be using our app on both mobile and tablet with different screen sizes.

1.3.2 Implementation Environment of the Current System

N/A The technological environment in which the product is to be installed is the iOS operating system for Apple products and the Android operating system for Android devices.

1.3.3 Partner or Collaborative Application

To run the software on a computer, the user will require the Flame library for game implementation. The software will not be expected to run if Flame is not installed on the User's computer.

For users who download the app on their smart device, no such installation is required.

1.3.4 Off-the-Shelf Software

- **Description:** The design will utilize the Flame library for game development.
Rationale: Flame is a very famous game building library in Flutter

and due to limitations in time and resources there is no need to reinvent the wheel.

- **Description:** The design will make use of the Flutter framework for the UI

Rationale: Flutter is a well documented framework that will make development of the UI much easier.

1.3.5 Anticipated workplace Environment

The anticipated workplace environment for Mastermind is everywhere. It is intended to be played in the house, on-the-go, public places etc. The application does not require access to mobile data or the internet hence it is accessible to all clients.

1.3.6 Schedule constraints

- ~~There are no set schedule constraints for our application however the documentation of the software will be delivered by the deadlines specified in the SE3XA3 outline.~~ The Proof of Concept Demonstration for Revision 0 is due week of February 14 2020
- The Test Plan for Revision 0 is due February 28 2020
- The Design & Documentation documents for Revision 0 are due March 13 2020
- The Revision 0 demonstration is the week of March 16 2020
- The Final Demonstration for Revision 1 is the week of March 30 2020
- The Final Documentation for the project is due April 6 2020
- ~~The unofficial deadline for mastermind's final implementation is set to April 1st 2020.~~

1.3.7 Budget constraint

Budget constraint is not applicable here as all the resources needed to complete the game have been provided to us.

1.4 Naming Conventions and Terminology

- **GUI:** graphical User Interface, the interface that the user interacts with that includes graphical elements, icons, and buttons.
- **Mastermind:** A two person game with pegs invented by Mordecia Meirowitz.
- **Flame:** A minimalist Flutter game engine library for creating easy to build games.
- **FPS:** Frames per second. A tool to measure the refresh speed of a image.
- **Dart:** A client optimized language for fast multi-platform app development.
- **Flutter:** Google's UI toolkit for crafting natively compiled applications for mobile development.
- **UI:** The interface which the user sees and interacts with. Similar to the GUI in this case.
- **Pegs:** What the user will see in an empty spot on the game board after making a guess. Pegs can be Red, Blue, Green, Yellow, White, Purple.
- **Hints:** Small tiles beside the game board that will tell the user how many of their guesses are in the correct position and are the correct colour.
- **Correct Guess:** This is the correct sequence of pegs that the user will need to input in order to win

1.5 Relevant Facts and Assumptions

1.5.1 Relevant Facts

- Flame is capable of hosting and running mastermind style games.
- Dart and Flutter are proven to be capable of running mobile games on both iOS and android.

- The original project has less than 700 lines of code in python.
- Mastermind is known to be one of the tools to determine a person's IQ.

1.5.2 Relevant Assumptions

The target user of the product is assumed to be:

- Mentally competent to play mastermind
- Physically capable of touching the screen with hand-eye coordination
- Interested in IQ evaluation methods

The average user will have :

- A relatively modern phone or tablet capable of running mastermind.
- Access to the Google Play Store or Apple App Store, as well as to the internet to download the game.

~~User characteristics should go under assumptions.~~

2 Functional Requirements

2.1 The Scope of the Work and the Product

2.1.1 The Context of the Work

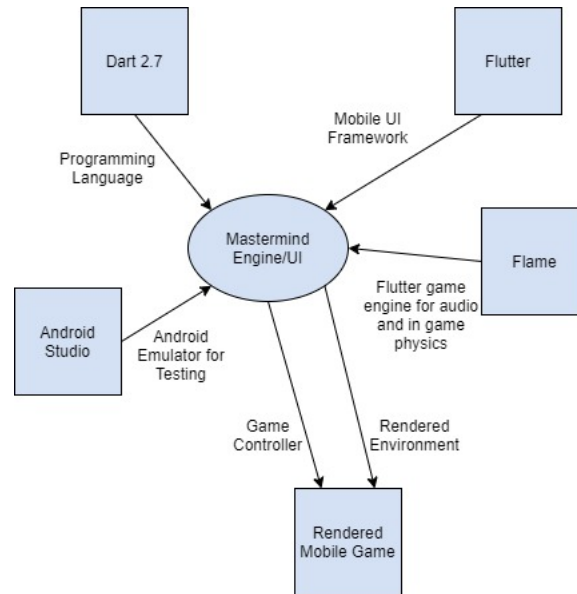


Figure 1: Context of the Work.

2.1.2 Work Partitioning

Table 2: Work Partitioning

Event Name	Input and Output	Summary
1. Mastermind Game Creation	Developer code (in)	Recreate mastermind game logic in Dart with same logic as original code base.
2. Mastermind Additional Level	Developer code (in)	Add additional challenges to the original mastermind game.
3. Player makes a guess	Player guess (in) Show correct guesses (out)	Update the state of board and allow user option to make more guesses
4. Game finishes	Final Player Guess(in) End of game message(out)	Update state of board to finished.

2.1.3 Individual Product Use Cases

The main use cases for this product allows for users to start a new game, make guesses, and view the rules of the game. ~~This case is~~ These cases are outlined in the diagram below.

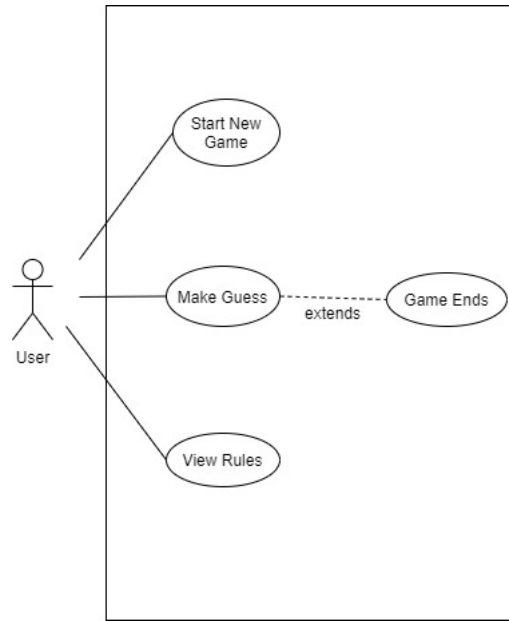


Figure 2: Use Case Diagram.

2.2 Functional Requirements

- **REQ1:** The game will be implemented using Dart and Flame library.
- **REQ2:** The user must be able to select a blue colour peg using the touch screen.
- **REQ3:** The user must be able to select a yellow colour peg using the touch screen.
- **REQ4:** The user must be able to select a red colour peg using the touch screen.
- **REQ5:** The user must be able to select a green colour peg using the touch screen.
- **REQ6:** The user must be able to select a white colour peg using the touch screen.
- **REQ7:** The user must be able to select a purple colour peg using the touch screen.

- **REQ8:** The user must be able to place the pegs onto the board using the touch screen.
- **REQ9:** ~~The user must be able to view the help screen using the touch screen.~~ The user must be able to view the instructions on how to play the game upon opening up the application.
- **REQ10:** The user must be able to close the game using the touch screen.
- **REQ11:** The user shall be able to select 4 pegs as a guess.
- **REQ12:** The system shall display hints to the user after each guess.
- **REQ13:** The correct sequence of pegs shall be displayed to the user after 10 incorrect guesses.
- **REQ14:** The user shall be able to undo a peg press if the row (guess) is not complete.

3 Non-Functional Requirements

3.1 Look and Feel Requirements

3.1.1 Appearance

- **LF1:** GUI should be visually appealing to users of our market audience. The colours representing buttons or game features should be easily distinguishable, and not blend with the background colour.
Fit Criterion: After the initial release of the project a survey should be completed by users to show that 95% of users enjoyed using the product.

3.1.2 Style

- **LF2:** The GUI should resemble the classic board from the game of Mastermind.
Fit Criterion: After the initial release of the project a survey should be completed by users to show that 95% of users thought the product resembled the classical version of Mastermind.

3.2 Usability and Humanity Requirements

3.2.1 Ease of Use Requirements

- **UH1:** The game shall be playable by people MIN_AGE and up.
Fit Criterion: A sample study of users MIN_AGE and up interacting with the game should show that 90% of them were able to play the game properly without additional assistance.
- **UH2:** The user shall be able to easily download and play the game on their smart device.
Fit Criterion: Testing will be done to ensure that there are no issues downloading the game from the Google Play Store or Apple App Store.
- **UH3:** The application shall be engaging and fun for users to play.
Fit Criterion: ~~Four months after~~ After the release of the product, ~~the ratings on the Google Play Store and Apple App Store should be above 4 (out of 5)~~ a sample of users will be asked to play the game, and after the experience 90% of the users should have said that they had fun and/or engaging..

3.2.2 Personalization and Internationalization Requirements

- **UH5:** The product shall allow users to select their preferred language.
Fit Criterion: A software test engineer will make sure that the users have the option to select their preferred language.

3.2.3 Learning Requirements

- **UH6:** Users of MIN_AGE and up shall be able to play the game with no prior experience or training.
Fit Criterion: A sample of users will be asked to play the game with no instructions. After PLAY_TIME minutes 90% of the users would have figured out how.
- **UH7:** Users shall be able to remember how to play the game after their first experience.
Fit Criterion: The same sample of users used in the criterion above will be asked back a week after their first session to test for memory

retention. 95% of said users should have remembered how to play the game.

- **UH8:** Instructions shall be provided for the rules of the game.
Fit Criterion: N/A

3.2.4 Accessibility Requirements

- **UH9:** The product shall use symbols and words that are naturally understandable by users.
Fit Criterion: Users will be invited in for an in-person assessment of symbols and words used in Mastermind. 95% of said symbols and words have to be deemed understandable by the group of users.
- **UH10:** Language used to describe the rules shall be easy to understand with no grammar or spelling errors.
Fit Criterion: A software test engineer will test to make sure that there are 0 spelling or grammatical errors in the application.
- **UH11:** The implementation of the product shall be hidden from the users.
Fit Criterion: Users should not be able to access or view any of the source code from within the app.

3.3 Performance Requirements

3.3.1 Speed and Latency Requirements

- **PR1:** The game interface shall respond to user input in under MIN_RESPONSE_TIME.
Fit Criterion: A software test engineer will perform a series of NUM_STRESS_TESTS stress tests to ensure that response time is always under MIN_RESPONSE_TIME. NUM_STRESS_TESTS divided by NUM_STRESS_TESTS₁₀₀/100 of the test cases shall pass.

3.3.2 Safety-Critical Requirements

- **PR2:** The product will not encourage violence, discrimination or cause addictive behaviour.
Fit Criterion: A survey will be conducted after the release of the product asking users if they have experienced an increase in any of the

aforementioned behaviours since starting to use the app. 98% of the users shall not have experienced any of the negative behaviours.

3.3.3 Precision or Accuracy Requirements

- **PR3:** User input shall translate directly to the intended output within the context of the game.
Fit Criterion: A software test engineer will perform ~~200~~ NUM_TEST_CASES test cases where the intended output is generated from the input 100% of the time.
- **PR4:** All colors displayed on the GUI shall be easily distinguishable.
Fit Criterion: A survey will be conducted on users asking about distinguish-ability of colours, 95% of users shall say that the colours were distinguishable.

3.3.4 Reliability and Availability Requirements

- **PR5:** The product shall be available for use UP_TIME.
Fit Criterion: The system will be tested once a day to ensure that there are no problems.
- **PR6:** The product shall achieve this UP_TIME, by using previous stable versions of the game for users during product maintenance.
Fit Criterion: N/A Throughout testing, a software test engineer will check in on the state of the game periodically throughout the game to make sure that it never goes down.

3.3.5 Robustness or Fault-Tolerance Requirements

- **PR7:** The product shall be playable without internet connection.
Fit Criterion: A software test engineer will play ~~100~~ MIN_NUM_TESTS iterations of the game with their device disconnected from the internet. They shall be able to successfully play 100% of the time.

3.3.6 Capacity Requirements

- **PR8:** The product shall be able to handle the loading of an individual user.
Fit Criterion: A software test engineer will play ~~100~~ MIN_NUM_TESTS

iterations of the game, with 100% success rate for completion of the game.

3.3.7 Scalability or Extensibility Requirements

- **PR9:** The product shall be implemented to allow for easy modification and improvement.
- **PR10:** The product shall allow for the implementation of new features.
Fit Criterion: The modularity of the software design will be verified by a third-party, by having them attempt to modify or add a feature to the game.

3.3.8 Longevity Requirements

- **PR11:** The product shall be expected to operate on all smart devices for the next several iOS and android system updates.
Fit Criterion: After the next major iOS or Android update, a survey will be given to users to gauge if their ability to play was impacted in anyway.

3.4 Operational and Environmental Requirements

3.4.1 Expected Physical Requirements

- **OE1:** The product shall be usable in any physical environment that a user can view their phone.
Fit Criterion: The project should be able to run any smart device with iOS or Android, or any computer with Flame downloaded.

3.4.2 Requirements for Interfacing with Adjacent Systems

- **OE2:** The product shall run on all smart devices running iOS or android operating systems.
Fit Criterion: Our application will be tested to run on several different devices to ensure compatibility.
- **OE3:** The product shall run using the most recent version of Dart (Dart 2.7).
Fit Criterion: N/A

3.4.3 Productization Requirements

- **OE4:** The product shall be able to be installed by an untrained user.
Fit Criterion: The product will be available on trusted platforms like the Apple app store or Google play store, where users have experience downloading trusted applications for their smart device.

3.4.4 Release Requirements

- **OE5:** The maintenance releases will not cause previous features to fail.
Fit Criterion: Maintenance releases will be thoroughly tested before release to ensure that all current working features are not impacted by new fixes/changes.

3.5 Maintainability and Support Requirements

3.5.1 Maintenance Requirements

- **MS1:** The product shall be updated when new bugs or glitches are known.
Fit Criterion: The developers will release a new patch every month to fix any bugs that have been made known by the community, or to release new content and features for the game.
- **MS2:** Maintenance releases shall be accompanied by patch updates to notify users of changes, new features or fixes.
Fit Criterion: N/A

3.5.2 Support Requirements

- **MS3:** Other programmers shall be allowed to identify and fix bugs for the open source product.
Fit Criterion: Contributions from other programmers will be validated before committing changes to the project.

3.6 Security Requirements

3.6.1 Access Requirements

- **SR1:** The product software will be free to download.
Fit Criterion: N/A

3.6.2 Integrity Requirements

- **SR2:** All data is stored on the user's local machine.
Fit Criterion: N/A
- **SR3:** The product shall protect itself from intentional abuse such as cheating.
Fit Criterion: Thorough testing will ensure that 99% of the data introduced to the game is safe from tampering that would allow players to cheat.

3.6.3 Privacy Requirements

- **SR4:** The product will not collect or store any user data.
Fit Criterion: The develops will not retain a single instance of personal information from users.

3.6.4 Audit Requirements

- N/A

3.6.5 Immunity Requirements

- **SR5:** All product download links shall be cleared of malicious software and viruses.
Fit Criterion: All download links issued by the development team will be thoroughly tested and secured to guarantee safe downloads.
- **SR6:** The product shall be installed only using trusted sources.
Fit Criterion: Provide a disclaimer to state that we are not responsible for download links not provided by our team.

3.7 Cultural and Political Requirements

- **CP1:** The product shall not discriminate any racial, political, cultural, or ethnic group.

Fit Criterion: Several months after the product release, a community poll will be conducted to gather feedback from the user community on any controversial subjects that they want to address.

3.8 Legal Requirements

- **LE1:** The product shall not violate any copyright or intellectual property laws.

Fit Criterion: Lawyers will review the product to ensure that the game does not violate any laws of copyright or stolen intellectual property.

3.9 Health and Safety Requirements

- **HS1:** The product shall not endanger or harm the user.

Fit Criterion: A test team will perform a risk analysis of the product to validate that it is safe for all users.

4 Project Issues

4.1 Open Issues

N/A

4.2 Off-the-Shelf Solutions

There currently exists many other free versions of Mastermind similar to the existing implementation. The common theme among all of these versions is that the graphical user interface is lacking in quality, and this is what our group aims to improve on.

These free versions can be used as a template to create the main logic for the basic version of the game. The benefit of using these other versions as templates is that it allows for our group to see where the current inefficiencies and trouble areas are, and work to fix these to improve the project.

Furthermore, there are many online resources for using Dart and Flutter to create 2-D games that our group will be able to utilize.

4.3 New Problems

4.3.1 Effects on the Current Environment

N/A The application should not take or use any data from the device that it is being played on. Furthermore, the application shall not alter or leave behind any malicious software on the device.

4.3.2 Effects on the Installed Systems

N/A

4.3.3 Potential User Problems

Any prolonged exposure to screens due to playing of Mastermind may potentially have adverse effects on the user's eyesight and eyes as a whole.

4.3.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

The development team has minimal experience making games using Dart and Flutter, so this may cause some issues when implementing some of the fine grain details of the user interface. The development team should focus on creating a simple and aesthetic version of the game first, before they decide to add some of the more intricate GUI details.

4.3.5 Follow-Up Problems

If there are updates to the Android operating system, the Dart/Flutter app may no longer be compatible. In this case maintenance would have to be done to ensure compatibility with the operating system.

4.4 Tasks

4.4.1 Project Planning/Development Process

Please refer to the [ProjectSchedule/ProjectSchedule.pdf](#) in the repository for a Gantt Chart detailing the tasks required for this project.

4.5 Migration to the New Product

4.5.1 Requirements for Migration to the New Product

N/A

4.5.2 Data That Has to be Modified or Translated For the New System

N/A

4.6 Risks

4.6.1 Excessive Schedule Pressure

Should the development team have any issues learning how to use Dart and Flutter then there is the potential to fall behind schedule as it would be very difficult to meet deadlines as well as learning the language/framework.

4.6.2 Management Malpractice

The development team will have to follow the plan outlined in the Gantt chart and ensure that proper use of time and resources is allocated in order to complete the project by the required due dates.

4.6.3 Low Quality

When developing games there is always the possibility of the team becoming too ambitious and focusing on the finer grained details rather than getting the main logic of the game working smoothly. This would result in a low quality game, thus letting the stakeholders down. To combat this risk the development team should focus on getting a basic implementation of the game working before any additional features are attempted

4.7 Costs

The developers will be making use of the open source languages; Dart and Flutter to create the product. Therefore there will be no costs associated with the development process.

4.8 User Documentation and Training

4.8.1 User Documentation Requirements

The development team will provide the user with an option to view the rules of the game from the start menu. The rules will dictate to the user how the game is to be played. Furthermore, inside of the game if a user makes an illegal move, the game will prohibit it and provide an explanation to the user on why the move was not allowed.

4.8.2 Training

There is no training required to use the product.

4.9 Waiting Room

~~At this time there is no plan to create additional functionality not already mentioned in the requirements.~~ some of the ideas that the team has for future additions to the game are:

- Adding varying levels of difficulty to the game. Currently the game requires the user to correctly guess a combination of 4 pegs, we could instead add an easier difficulty where they only have to guess 3 pegs, and likewise a more difficult setting where they have to try and guess 5 pegs.
- Adding support for multiple languages (French, Spanish, etc.)
- Create a desktop version of the application.
- Create a "Play a Friend" mode in which two friends will be able to play MasterMind over a server.
- Create a leader board in which the user will be able to see their all time win percentage, as well as their friends statistics.

4.10 Ideas for Solutions

The team plans to user proper documentation, code hierarchy, and testing. Alternatively, the team could use a different combination of languages/graphics libraries such as python and pygame for the implementation.

5 Appendix

5.1 Symbolic Parameters

The following are definitions of symbolic constants that are used throughout this document:

- MIN_AGE: 4
- MIN_RESPONSE_TIME: 0.5 Seconds
- UP_TIME: 24 hours a day, 365 days a year
- **PLAY_TIME: 10 Minutes**
- **NUM_STRESS_TEST: 100**
- **NUM_TEST_CASES: 200**
- **MIN_NUM_TESTS: 100**