[INITIAL DESIGN REPORT]

MasterMind
## Contents

1. Introduction .................................................................................................................. 4  
   1.1. Problem Definition ............................................................................................... 4  
   1.2. Purpose of the Project ......................................................................................... 4  
   1.3. Scope of the Project ............................................................................................. 4  
   1.4. Overview ............................................................................................................... 5  
   1.5. Definitions and Abbreviations .............................................................................. 5  
   1.6. References ............................................................................................................ 6  
2. System Overview .......................................................................................................... 6  
3. Design Considerations .................................................................................................. 6  
   3.1. Design Assumptions, Dependencies, Constraints ................................................. 6  
      3.1.1. Hardware Constraints ................................................................................... 7  
      3.1.2. Software Constraints ................................................................................... 7  
      3.1.3. Time Constraints ......................................................................................... 8  
      3.1.4. Security Constraints ................................................................................... 8  
   3.2. Design Goals and Guidelines .............................................................................. 8  
      3.2.1. Portability ..................................................................................................... 8  
      3.2.2. Speed Considerations ................................................................................. 9  
      3.2.3. Reliability .................................................................................................... 9  
      3.2.4. User Interactivity ....................................................................................... 9  
4. Data Design .................................................................................................................. 10  
   4.1. Data Description ................................................................................................... 10  
   4.2. Data Dictionary ................................................................................................... 11  
5. System architecture ...................................................................................................... 12  
   5.1. Architectural Design ............................................................................................ 12  
   5.2. Description of component .................................................................................. 16  
      5.2.1. Server ........................................................................................................ 16  
      5.2.2. Game .......................................................................................................... 18  
      5.2.3. Emotiv ......................................................................................................... 22  
   5.3. Design rationale .................................................................................................... 28  
6. User Interface Design ................................................................................................. 28  
   6.1. Overview of User Interface .................................................................................. 28  
   6.2. Screen Images .................................................................................................... 29  
      6.2.1. Welcome Screen .......................................................................................... 29
1. Introduction

1.1. Problem Definition

Many people, especially the children, the elderly, the sportsman and the businessmen are suffering from attention deficit disorder and concentration problems. Such disorders are “characterized by poor attention and distractibility and/or hyperactive and impulsive behaviors”. It is one of the most common mental problems children are faced with, assumed approximately 3-5% being affected. There are many medical and therapeutic treatments offered to this problem. Still, none can capture all of the groups of patients. Especially for children, an effective and attractive solution has not been fully developed yet.

1.2. Purpose of the Project

Stemming from the lack of an attractive solution to the problem mentioned in the above section, the purpose of current project is to develop a software product that would provide a solution to the treatment of such individuals. It is aimed to be done via a computer game that utilizes the Emotiv EPOC device. This device will collect data regarding the patient’s attention and concentration level as he/she plays the computer game.

1.3. Scope of the Project

In developing this computer game, it is planned to make use of a mechanism, namely the neuro-feedback mechanism, which enables controlling the nervous system activity of the patients. The users/patients will be presented with a computer game throughout which they
will concentrate on certain stimuli. The Emotiv device will record the changes in the concentration level of the patient and transmit it to the computer. If the patient’s concentration decreases at a point, the game requirements will increase in response to this, which will serve as neuro-feedback to the patient. In effect, the patient will be forced to improve his/her management skills in order to win the game. By these means, the product will help both the diagnostic and the treatment processes through these means. In addition, patients will be able to see brainwaves feedback from computer screen and manage to control their brainwaves by audio - visual feedbacks coming from the system.

1.4. Overview

This document captures the software components, interfaces, and data that would be required for the implementation of our product. Accordingly, a general overview of the project and the assumptions, constraints, and goals of the design will be put forth initially. Data design, the architecture of the program and the information regarding the user interface will follow. A detailed design then will be given which explains any specifications of the components mentioned before. Finally, the libraries and tools to be used will be explained and a time plan will be provided.

1.5. Definitions and Abbreviations

BCID: Brain Computer Interface Devices

EEG: Electroencephalography is the recording of electrical activity along the scalp produced by the firing of neurons within the brain

Neuro-feedback: A mechanism that enables illustrating brain activity
Brain-waves: A rhythmic fluctuation of electric potential between parts of the brain,

Emotiv EPOC: An EEG-based BCI device.

1.6. References

- Minder IT Epoclib library training

2. System Overview

This software product will serve as a way of treatment for patients with attention deficit, by providing them with a computer game that has neuro-feedback mechanism. In accordance with this mechanism, a table will appear on one side of the screen which shows the patient his/her own concentration level, and thereby urges the players to concentrate their attention on specific stimuli in the game. As the patients play the game, their alpha and beta waves will be recorded and this data, recorded by the Emotiv EPOC device, will be displayed as a report at the end of the game. The report will be accessible to the doctors for further use in the treatment process.

3. Design Considerations

3.1. Design Assumptions, Dependencies, Constraints

We are planning to apply the Spiral Methodology throughout the design and development processes of our software product. The methodology can be summarized as in the diagram below:
3.1.1. Hardware Constraints

We will make use of Emotiv EPOC device. This will also require a wireless USB connection and a computer with at least average properties.

3.1.2. Software Constraints

A Windows or Linux operating system will be needed as well as Adobe Flash Player in order for the patients to display and play the game.
3.1.3. Time Constraints

As mentioned before, this game is prepared in an attempt to help patients with attention deficit. Therefore, the game should not be any longer than 5 minutes, so that the players can fully focus their concentration on the game. Moreover, the game should be longer than 1 minute to provide the players with sufficient time to understand and get used to the game’s requirements.

3.1.4. Security Constraints

Keeping all the information gathered from the patients’ performance and not disclosing them to anyone is a crucial issue that will be taken into consideration. Thus, all information will be stored in our own database and the profiles will be accessible only to experts or doctors who have permission to reach. They will be able to reach the profiles using their own ids and passwords.

3.2. Design Goals and Guidelines

3.2.1. Portability

Since we will make use of Java, the software product will be portable to any computer with average properties. The product will be accessed only from hospitals and clinics.
3.2.2. Speed Considerations

We will be recording the alpha and beta values of patients’ measures in every 250 milliseconds, a feature provided by the Java library. This can be considered as a bottleneck of the product; because even if the ball’s speed during the Pong game increases more, the program will be able to record the responses in every 250 milliseconds, not any more frequently.

3.2.3. Reliability

We will be designing a fault tolerant software product. Since each and every of its recordings may not be all true, it will accumulate the recordings to reach the true result. As mentioned above, the recordings will be done in every 250 milliseconds. We are going to store all of them in our own database. Profiles will only be accessed by doctors and they are not accessible for any other user. This means that in a 4-minute record, the alpha and beta values will be recorded for 16 times. The machine will collect these measures and approximate the true values as the number of games played by the same patient increases.

3.2.4. User Interactivity

We are planning to design a pretty user-friendly and easy-to-use product, since one of the main targets of this product is children with attention deficit problems. As is shown in the picture on section 6, the interface will consist of two parts: one of which is the screen where the game is displayed and the second is the place where the player’s performance is shown as feedback to the player himself. Moreover, special emphasis will be given so that there is no distracter on the screen that would probably attract the player’s attention away from the game.
4. Data Design

4.1. Data Description
### 4.2. Data Dictionary

<table>
<thead>
<tr>
<th>Name</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>User stands for the user of the software product, in our case for the patient. Its primary key is User Id which differentiates one user from the others. Other than this every user has a name which stores the name of the user, a password which authorizes user, and an e-mail part for communication reasons.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Game</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Game stands for the Game part our software product. Its primary key is Game Id which differentiates one game from the others, because we are planning to make different games. Every game will be logged according its date, when it is played. Game will have also a name part, indicating its name.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Emotiv Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Emotiv Data stands for our Log part of our Software. Its primary key is Data Id which differentiates one users data from the others, since all datas will be logged according their users. Date, User Id and Game will be parts of Emotiv Data.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Expert</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Expert stands for the Psychology Expert who is responsible from treatment of the patient/user. Its primary key is Expert Id which differentiates one expert from the others. Every expert will responsible for some patients according their User Id’s. Experts will have passwords which allow them to reach the logs of the patients. Other than this this part includes, the name of the expert; name and the e-mail adress of the Expert; Email.</td>
</tr>
</tbody>
</table>
5. System architecture

In this section of document, program architecture and system components will be presented.

5.1. Architectural Design

Our Project has three main components:

- Server Component
- Game Component
- Emotiv Component

Briefly, game component works on pc which Emotiv and server components are connected to it.

To illustrate component diagram is prepared.
Emotiv device operate as input source for game component. It collects data from user and transmits them into the pc. At this point Game component operations begin. While user is playing game, data is acquired from Emotiv device. After game play changes are applied reaction of user will be observed. Also, Game component was in contact with server component entire game time. It sent all necessary kind of data to server.

To better understanding of component operations, Level-0 Data Flow Diagram is prepared:

**Level-0 Data-flow Diagram**

Server Components is the database server of game. It handles keeping and monitoring acquired data from users. It keeps data from user and allows experts to reach them. It stores logs about user info, brainwaves, game. Game component send those logs during game.
Emotiv Component is responsible collecting brainwave changes and transmitting them to pc. Also, it manages transformations of brainwaves into usable data.

Game Component is main component. Other components are linked to game component. It implements data from Emotiv device, enables to experts implement log operations via server component and provide user with game.

Data-flow of program makes easy to understand architecture of software. For this reason Level-1 Data-flow diagram is prepared:
5.2. Description of component

As it mentioned before our Project has three main components:

- Server Component
- Game Component
- Emotiv Component

5.2.1. Server

Server component is database of our program. It will responsible receiving log files (they will come in XML format), updating database and return experts needed log files.

5.2.1.1. Processing narrative for Server

MasterMind Project is designed for helping people who are suffering attention deficit disorder and concentration problems. By providing psychology experts’ additional inputs and data from patients (users) it is aimed those patients (users) will get better treatment options. Experts may treat better with new types of data that we provide for them. In Game part, there are several of data type are stored to database. Server part enables experts to reach patient data efficiently. By evaluating them they will also help their patients more efficiently.
5.2.1.2 Server interface description

Receive:

<table>
<thead>
<tr>
<th>xmlFile, FileStream</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ setConnection(): void</td>
</tr>
<tr>
<td>+ receiveXML(): void</td>
</tr>
<tr>
<td>+ parseXML(): void</td>
</tr>
<tr>
<td>+ updateDatabase(record: string): void</td>
</tr>
</tbody>
</table>

Database Connector:

| + connectDatabase(): void |
| + updateUserInfo(record: string): void |

5.2.1.3 Server processing detail

Receive:

- void setConnection(): This method will set up required database connection.
- void receiveXML(): Receive XML file sent by game.
- void parseXML(): Parse received XML file for updating database.

DatabaseConnector:

- void connectDatabase(): Send connection request.
5.2.1.4 Dynamic behavior of Server

5.2.2. Game

Game is the part requires all inputs. Also, all implementations are made in this component. User plays the game while game and Emotiv data are transferred to database.

5.2.2.1 Processing narrative for Game

People who are suffering attention deficit disorder and concentration problems are mostly children. Reaching them is not always easy thing. Without reaching and communicating
them, solving their problem is difficult task because gathering information about them will be troubling for psychologist. At this point our solution will come up. We thought that if child does not aware of s/he is being examined, better information can be gathered. While s/he is playing game, difficulty changes depending on his/her concentration level and monitoring his/her reactions basic functions game will apply. On the other hand, if user is not children but adult, psychologist gets additional information about him/her other than classical approach. Either way it is useful and brings new perspective to classical treatment ways.

5.2.2.2. Game interface description

[Diagram of user interface, game, database connector with methods listed]
5.2.2.3. Game processing detail

- **void login(username, password, mode):** User login the system with his/her username and password. Mode determines user’s role. If user is psychology experts s/he is able to reach database and logs. Otherwise, user can only play game.
- **Game gameSelect(gameid):** It enables chosen game.
- **void getUserInfo(userid):** If user’s mode is psychologist s/he can retrieve data from database server.
- **void startGame():** Starts chosen game.
- **void exit():** Exit current game.
- **void connectDatabase():** Send connection request.
- **void updateUserInfo(record: string):** Update users information.
5.2.2.4. Dynamic behavior of Game
5.2.3. Emotiv

Emotiv is the device which is responsible for collecting data from user. Those data will be used in game component. Hence, only task of Emotiv component is collecting brainwaves from user and transmitting them.

5.2.3.1. Processing narrative for Emotiv

It is known that brainwaves are changed according to person’s psychological status. Also those waves can be measured by special devices such as Emotiv. Emotiv collect brainwaves as raw data from user, and then by implementing Fourier Transformation it makes those raw data into usable band data. Band data can be used several ways but in MasterMind project they are going to use user concentration levels and attention status.

5.2.3.2. Emotiv interface description

```
EpocLib

public boolean connectToBCI();
public void addEEGRawDataListener(EEGRawDataListener, long);
public void removeEEGRawDataListener(EEGRawDataListener);
public void addEEGBandDataListener(EEGBandDataListener, long);
public void removeEEGBandDataListener(EEGBandDataListener);
public EEGSampleBandData getEEGSampleBandData(long duration);
public void exitApplication();
```
5.2.3.3. Emotiv processing detail

Emotiv component does its job via these methods:

- public boolean `connectToBCI()`: After Emotiv is connected and program started this method starts two new threads. First one is started for reading and the other one is started for writing raw EEG data.

- public void `addEEGRawDataListener(EEGRawDataListener, long)`: In a given period (in milliseconds) `addEEGRawDataListener` method enables gathering data from user. This method is used for raw data listening.

- public void `removeEEGRawDataListener(EEGRawDataListener)`: Disables raw data listener.

- public void `addEEGBandDataListener(EEGBandDataListener, long)`: In a given period (in milliseconds) `addEEGBandDataListener` method enables gathering data from user. This method is used for band data listening.

- public void `removeEEGBandDataListener(EEGBandDataListener)`: Disables raw data listener.

- public `EEGSampleBandData getEEGSampleBandData(long duration)`: In a given period (in millisecond) this method get sample band data from user.

- public void `exitApplication()`: By ending threads it terminates data flow.
5.2.3.4. Dynamic behavior of Emotiv connectToBCI():
5.3. Design rationale

Designing any project as one piece is troublesome way for implementation and teamwork. Dividing project into sub-components makes implementing phase easier. Developer’s focus will be high and s/he is concerned smaller problems. While dividing project, components’ structural preferences are taken into consideration. MasterMind Project requires Emotiv device, database application and game application. Since these three applications are independent from each other, it would be logical way to divide into three sub-components. Their independency provide us smaller target for focusing. We could concentrate on easier problems and solve them. After finishing components, only combining phase will be remain.

6. User Interface Design

6.1. Overview of User Interface

The software will be available for two kinds of users: Players (users) and Experts. Therefore, the user needs to change his/her role at the beginning of the program. Users will be also granted username and password in order to keep privacy. That is, users’ data will be stored in a database with protection and experts can only see and examine their own patients.

- **Player:** After a player logs in, s/he will see a page to choose a game. For the first term, our team decided to create Pong game. The user will play this game with standard input devices, but the speed of the game will depend on the user’s concentration level. While user is playing the game, the data coming from Emotiv
Device will be interpreted and affect the game and these data’s will be sent to the database.

- **Expert:** After an expert logs in, s/he will see the patients (users) that are related with him/her. When the expert selects a patient, the user’s log will be displayed and the expert can see when the user played which game, what is his/her score and most important of all, his/her brainwave data’s coming from Emotiv Device which helps him/her treat patients.

**6.2. Screen Images**

As for now, we are at very beginning of our software. Therefore, most of the functions we mentioned have not been implemented yet.

**6.2.1. Welcome Screen**
6.2.2. User Game Selection Screen

![User Game Selection Screen](image1)

6.2.3. Game Screen

![Game Screen](image2)
6.3. Screen Objects and Actions

Most of the software consists of buttons, labels and text lines and their actions can easily be determined.

In the game part, the screen generally consists of three main parts. The line chart will show player’s concentration level. Data of this graph will be provided from interpreted Emotiv Device data. Also, user’s change of concentration will affect the outcome of the game. For example, the loss of concentration will increase the speed of the ball for this pong game. Main aim here is to keep player motivated all the time. This game will last for a limited time and player is able to see remaining time from time part. There will be a counter in this part. Time of the game will be decided after we examine a good treatment time for these games.

7. Detailed Design

7.1. Server Component

Classification: Component

Definition: database of the software

Responsibilities: responsible receiving log files, updating database and return experts needed log files

Constraints: User needs to have internet connection to access server. Also, the user must have a username and a password to log in to server and update data.
**Composition:** Database connection to our server will be satisfied by using Database Connector class. With given username and password, the class will connect the database and it will have authority to update user information.

Incoming data from database will be handled by Receive class. With given parameters, the class gets data from database. Since the structure of the database based on XML file system, the class will also parse the data and get required information.

**Uses/Interactions:** The database will be in an interaction with the software (game). As it can be inferred from the component diagram in page 12, the saved game data will be stored in database. Also, the other kind of users which is experts can receive data by using the software.

**Resources:** As written before, the main source of this component is database. Any conditions and/or deadlock situations will be resolved according to the type of the database. That is, the main methods of the specified database will be used to solve problems.

**Processing:** Since it is based on database, it must be reachable by users all the time. Also, it must be reachable anywhere. If any exceptional conditions occur, we will resolve using the standard backup processes of databases.

### 7.2. Game Component

**Classification:** Component

**Definition:** game of the software
**Responsibilities:** responsible for Emotiv data transferring to database while the user is playing the game.

**Constraints:** User needs to have internet connection to access server. Also, the user must have a username and a password to log in to server and update data. In addition, user must have Emotiv device to play the game. Some inputs of the game component are based on Emotiv data coming from the Emotiv neuroheadset.

**Composition:** User is asked to enter username and password in the user interface. After that the Database Connector will look to the database and check whether the user is valid or not. Then the user is asked to select a game.

By selecting the game, game class will be called with chosen id and game will start. After collecting data, the game will connect to the database and update the user’s information with interpreted data.

**Uses/Interactions:** The game will be in an interaction with the database and Emotiv device.

As it can be inferred from the component diagram in page 12, the saved interpreted game data will be stored in database. Also, the data coming from Emotiv neuroheadset will be received and interpreted. The interpreted data will affect the game.

**Resources:** Standard java libraries will be used to create user interface and the game. Also, epoclib library will be used to communicate with the brain control interface device. The raw data will be coming from the device and epoclib library will be used again to interpret data. These data will send to the game and affection of the result will be shown to the user. At the same time it will send to the database.
**Processing:** Since it is based on the brain control interface device, the game will not start without this device. User must satisfy the conditions to play the game. Also, s/he needs to have username and password to enter the game. After conditions are satisfied, the game will be initialized by the software. If any exceptional conditions occur, the software will start to look the conditions and it will not initialize the game until they are satisfied again.

### 7.3. Emotiv Component

**Classification:** Component

**Definition:** interface of the brain control interface device

**Responsibilities:** responsible for collecting data from user

**Constraints:** In addition, user must have Emotiv device to play the game. Also, all of the receptors need to be working and placed correctly. Some inputs of the game component are based on Emotiv data coming from the Emotiv neuroheadset.

**Composition:** First of all, the software will connect to the brain control interface device by using epoclib library. After that, the software will check all of the receptors are working and signal strengths are efficient. With conditions satisfied, the software will receive data from the device and interpret them.

**Uses/Interactions:** The Emotiv component will be in an interaction with the game. Most of the game inputs is handled by using device connection.

**Resources:** epoclib library will be used to communicate with the brain control interface device. The raw data will be coming from the device and epoclib library will be used again to
interpret data. These data will send to the game and affection of the result will be shown to the user. Within this timeline, some libraries coming from the Emotiv SDK will be used to check signal efficiency of the receptors.

**Processing:** Since it is based on the brain control interface device, the software will not allow the user to continue to game. After conditions satisfied, EEGDataListener will be initialized to collect data. Within this period, if any exceptional conditions occur, the software will warn the user. Also, the software reset itself because the game data without the Emotiv data is useless.

8. Libraries and Tools

Since we planned to implement this software in Java language, we need standard Java libraries and some other libraries to connect the Brain Computer Interface Device and receive data. Other libraries are listed below.

8.1. Libraries

8.1.1. Epoclib Library

This library is provided by our sponsor company “Minder IT”. This library includes some methods to connect Brain Computer Interface Device (BCID) and acquire the user’s raw and interpreted brainwave data. It is suitable for Emotiv Device so it is one of the best options for our team.
8.2. Tools

8.2.1. Emotiv Software Development Kit

The Emotiv Software Development Kit includes a high resolution, neuro-signal acquisition and processing wireless neuroheadset and Emotiv’s proprietary software toolkit that exposes Emotiv APIs and detection libraries.

8.2.2. NetBeans

The NetBeans IDE is an open-source integrated development environment. NetBeans IDE supports development of our Java application type.

9. Term Planning
9.1. Term 1 Gantt Chart
## Gantt Chart for Term-2

<table>
<thead>
<tr>
<th>Number</th>
<th>Task</th>
<th>Resource</th>
<th>Start</th>
<th>End</th>
<th>Duration</th>
<th>Q1 - 2011</th>
<th>Q2 - 2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Implementation of first game</td>
<td></td>
<td>10/2/2011</td>
<td>10/3/2011</td>
<td>21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Improvements in the first game</td>
<td></td>
<td>20/3/2011</td>
<td>30/3/2011</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Improvements in the second game</td>
<td></td>
<td>20/4/2011</td>
<td>20/4/2011</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Tests for the third game</td>
<td></td>
<td>5/5/2011</td>
<td>10/5/2011</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Improvements in the third game</td>
<td></td>
<td>10/5/2011</td>
<td>15/5/2011</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Finalization of Database</td>
<td></td>
<td>1/5/2011</td>
<td>25/5/2011</td>
<td>18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Finalization of Software</td>
<td></td>
<td>25/5/2011</td>
<td>30/5/2011</td>
<td>4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
10. Conclusion

10.1. Finished Work

We decided that we should develop a single Pong game our application and at the demo time, at the end of the semester, we will add some other games.

Until now, we set up the Emotiv SDK, NetBeans IDE and other necessary environment. We try to figure out how a desktop application is implemented and develop small Java Desktop applications. For example, we developed a small application which connects to Emotiv neuroheadset and receive data. Moreover, we implemented another web application which can parse an XML file and display on the screen.

In addition, for the server and web page module, we set up MySQL, and Apache. We examined how they work and developed a small part of our project. In detailed, we basically construct our database, and connected entities to each other using primary and foreign keys.

10.2. Future Work

The main flow of the project can be examined in the Gantt chart section. In this section, near future works for the prototype implementation will be discussed. In the following weeks, till the day which the demo will take place; we will be working on the prototype. We will improve a small pong-game and make it suitable and reliable in terms of interpretation of brainwaves to the demo. Implementing this first prototype, we will certainly face with some problems; those will help us to clarify the final construction elements of our system. The necessary changes will be done and the details of the classes, sequence etc. will be
determined. Later after the prototype, we will be focusing to the details of the neuro-feedback and complexity to the games. This part is also important to reach and interact with the users. Debugging and reimplementaion will be done to maintain a consistence and stability for our system. All these works will help us to see problems may appear and to get experience on the project. After all, the final design report will be written that will provide us to start implementing the project in the second term.