

Version	Description of Change	Author	Date
1	Creation	Team Hygge	16-10-19
2	Milestone 3 Update	Team Hygge	7-11-19

## CONTENTS

1	INTRODUCTION.....	4
1.1	Purpose.....	4
1.2	Scope.....	4
1.3	Background.....	4
1.4	References.....	4
1.5	Assumptions and Constraints.....	4
1.6	Document Overview.....	5
2	METHODOLOGY.....	5
3	FUNCTIONAL REQUIREMENTS.....	5
4.1	Context.....	5
4.2	User Requirements.....	5

4.3 Data Flow Diagrams.....	6
4.4 Logical Data Model/Data Dictionary.....	6
4.5 Functional Requirements.....	6
5 OTHER REQUIREMENTS.....	6
5.1 Interface Requirements.....	6
5.2 Data Conversion Requirements.....	7
5.3 Hardware/Software Requirements.....	7
5.4 Operational Requirements.....	7
APPENDIX A - GLOSSARY.....	
11	

# **1 INTRODUCTION**

We are striving to create a website where students can come to relax and forget about their problems for a while. The website will be a place where students can come to see some funny videos for a good laugh, or find some guided meditations for those moments of anxiety, or find some good study music, or even be redirected to get some real time help if struggling with something more major.

## **1.1 Purpose**

The purpose of this project is to help contribute to better mental health for students. We are aware of the high rate of stressed out students and high rates of suicide, and we really wanted to help students that may be struggling through this project.

## **1.2 Scope**

This document will be a continuously changing document that will evolve and grow along with our project. As we progress with our project, this document will begin representing all of our ideas, accomplishments, changes and in the end provide a thorough breakdown on why and how our website was created.

## **1.3 Background**

Hygge (pronounced “Hoo-gah”) is a website made for students by students. The purpose of this website is to allow students an uncomplicated, fun place where they can go to relax and forget about their problems for a little while.

This document will be produced by the entire team in order to divide the work of the project fairly and allow team members that worked on specific parts of the project to explain it in their own words.

## **1.4 References**

[List references and controlling documents, including: meeting summaries, white papers, other deliverables, etc.]

## **1.5 Assumptions and Constraints**

### **1.5.1 Assumptions**

We are assuming that there will be no copyright issues with the music, and guided meditations we intend on using. We are assuming that we can utilize the Hercules server for our website and that Eclipse will be a sufficient tool to complete our website.

### **1.5.2 Constraints**

The constraints that we face as a team:

- Limited experience in creating websites ( have only taken one class regarding that, CS 215)
- We have a short amount of time to create our website while managing our other classes, projects, commitments.

## **1.6 Document Overview**

[Provide a description of the document organization.]

# **2 METHODOLOGY**

The FRD document has been created based on the template given by our professor, Tim Maciag. The modelling methods used in this project are very straightforward and are explained thoroughly in their respective sections.

### 3 FUNCTIONAL REQUIREMENTS

#### 4.1 Context

The following is our Model View Controller (MVC) and the class diagram that describe our website.

Figure 1: Model View Controller

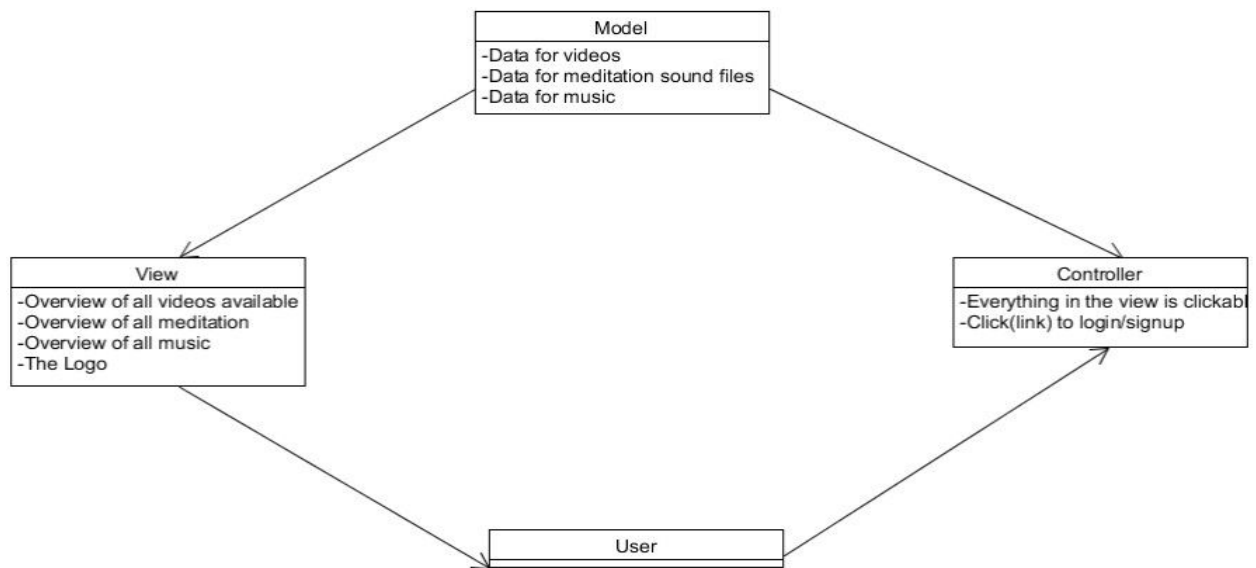
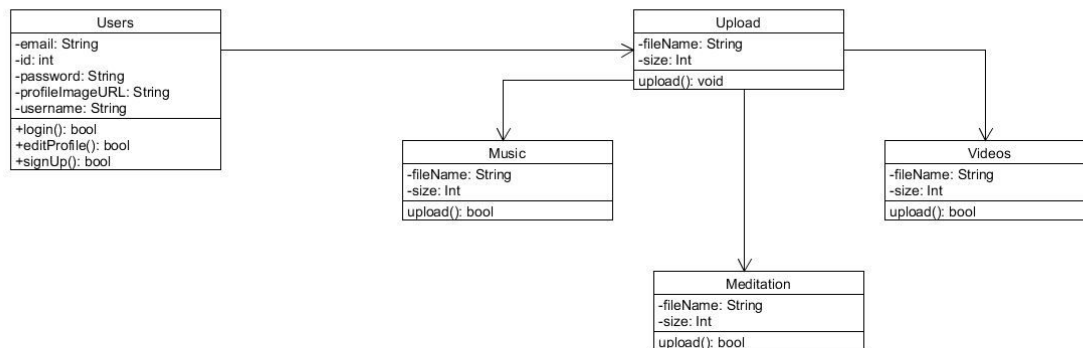


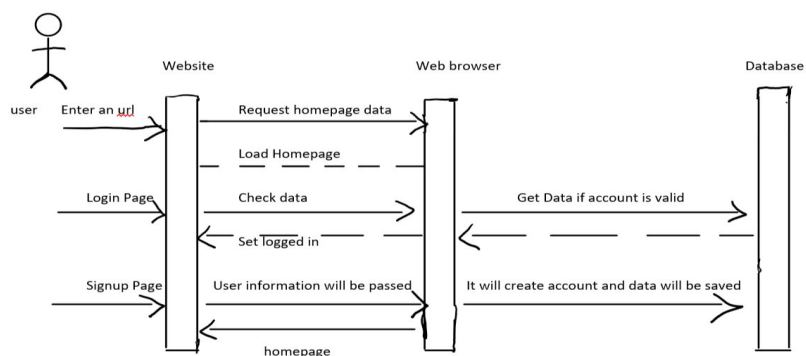
Figure 2: Class Diagram



## 4.2 User Requirements

The following is our sequence diagram that shows how the user's actions interact with the website. To explain the sequence diagram, we start off from the homepage of our website which contains the links to all our content and our login and sign up pages. Once a user clicks on a link or requests to login or signup, the request goes to the web browser which then requests that data from the database. If the data exists, it will be given back to the web browser which will then display it on the website.

Figure 3: Sequence Diagram



## 4.3 Data Flow Diagrams

[Decompose the context level diagrams to determine the functional requirements. Data flow diagrams should be decomposed down to the functional primitive level. These diagrams are further decomposed during design.]

## 4.4 Logical Data Model/Data Dictionary

[Create the initial Logical Data Model. Describe data requirements by providing data entities, decomposition, and definitions in a data dictionary. The data requirements describe the business data needed by the application system. Data requirements do not describe the physical database and are not at the level of identifying field names.]

## 4.5 Functional Requirements

[List the functional requirements of the system.]

### 4.5.1 Functional Requirements Group1

[List the functional requirements for each functional requirements group.]

Exhibit 4 - Sample Requirements Group 1

Section/ Requirement ID	Requirement Definition
FR1.0.	The system shall [parent requirement group 1].
FR1.1	The system shall [child/parent requirement].
FR1.1.1	The system shall [child requirement].
FR1.1.2	The system shall [child requirement].

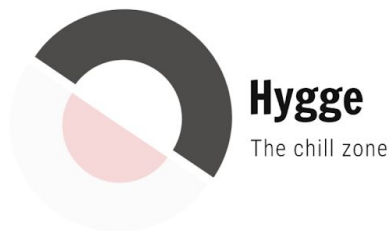
### 4.5.2 Functional Requirements Group 2, Etc.

# 5 OTHER REQUIREMENTS

[Describe the non-behavioral requirements.]

## 5.1 Interface Requirements

HOMEPAGE:





## LOGIN PAGE:

# Welcome Back

Login

[Logout](#)

## SIGNUP PAGE:

Signup Form

Enter Email

Enter Username

Enter Password

Retype Password

SignUp

[Go Back](#)

### 5.1.1 Hardware Interfaces

[Define hardware interfaces supported by the system, including logical structure, physical addresses, and expected behavior.]

### 5.1.2 Software Interfaces

[Name the applications with which the subject application must interface. State the following for each such application: name of application, external owner of application, interface details (only if determined by the other application).]

It is acceptable to reference an interface control document for details of the interface interactions.]

### **5.1.3 Communications Interfaces**

[Describe communications interfaces to other systems or devices, such as local area networks.]

## **5.2 Data Conversion Requirements**

Not applicable for Project Hygge.

[Describe the requirements needed for conversion of legacy data into the system.]

## **5.3 Hardware/Software Requirements**

We are using Hercules as our database. To code our website, we used Notepad++ and saved our files with the appropriate extensions and updated that to the database.

[Provide a description of the hardware and software platforms needed to support the system.]

## **5.4 Operational Requirements**

[Provide the operational requirements in this section.]

Do not state how these requirements will be satisfied. For example, in the Reliability section, answer the question, “How reliable must the system be”? Do not state what steps will be taken to provide reliability.

Distinguish preferences from requirements. Requirements are based on business needs, preferences are not. If, for example, the user requires a special response but does not have a business-related reason for it, that requirement is a preference.

Other applicable requirements on system attributes may be added to the list of subsections below.]

Operational requirements describe how the system will run and communicate with operations personnel.

### **5.4.1 Security and Privacy**

[Provide a list of the security requirements using the following criteria:

- A. State the consequences of the following breaches of security in the subject application:

1. Loss or corruption of data
  2. Disclosure of secrets or sensitive information
  3. Disclosure of privileged/privacy information about individuals
  4. Corruption of software or introduction of malware, such as viruses
- B. State the type(s) of security required. Include the need for the following as appropriate:
1. Physical security.
  2. Access by user role or types.
  3. State access control requirements by data attribute. For example, one group of users has permission to view an attribute but not update it while another group of users has permissions to update or view it.
  4. State access requirements based on system function. For example, if there is a need to grant access to certain system functions to one group of users, but not to another. For example, "The system shall make Function X available to the System Administrator only".
  5. State if there is a need for certification and accreditation of the security measures adopted for this application]

*The Security Section describes the need to control access to the data. This includes controlling who may view and alter application data.*

#### **5.4.2 Audit Trail**

[List the activities recorded in the application's audit trail. For each activity, list the data recorded.]

#### **5.4.3 Reliability**

- A. [State the following in this section:
1. State the damage can result from failure of this system—indicate the criticality of the software, such as:
    - a) Loss of human life
    - b) Complete or partial loss of the ability to perform a mission-critical function
    - c) Loss of revenue
    - d) Loss of employee productivity
  2. What is the minimum acceptable level of reliability?
- B. State required reliability:

1. Mean-Time-Between-Failure is the number of time units the system is operable before the first failure occurs.
2. Mean-Time-To-Failure is the number of time units before the system is operable divided by the number of failures during the time period.
3. Mean-Time-To-Repair is the number of time units required to perform system repair divided by the number of repairs during the time period.]

*Reliability is the probability that the system processes work correctly and completely without being aborted.*

#### **5.4.4 Recoverability**

We do not have enough time in the semester to consider all of these scenarios. If we had a longer amount of time on this project, we would be able to consider these questions and plan for them as well.

- A. In the event the application is unavailable to users (down) because of a system failure, how soon after the failure is detected must function be restored?
- B. In the event the database is corrupted, to what level of currency must it be restored? For example "The database must be capable of being restored to its condition of no more than 1 hour before the corruption occurred.
- C. If the processing site (hardware, data, and onsite backup) is destroyed, how soon must the application be able to be restored?]

*Recoverability is the ability to restore function and data in the event of a failure.*

#### **5.4.5 System Availability**

[State the period during which the application must be available to users. For example, "The application must be available to users Monday through Friday between the hours of 6:30 a.m. and 5:30 p.m. EST. If the application must be available to users in more than one time zone, state the earliest start time and the latest stop time. Consider daylight savings time, too.

Include use peak times. These are times when system unavailability is least acceptable.]

*System availability is the time when the application must be available for use. Required system availability is used in determining when maintenance may be performed.*

#### **5.4.6 General Performance**

[Describe the requirements for the following:

- A. Response time for queries and updates
- B. Throughput
- C. Expected rate of user activity (for example, number of transactions per hour, day, or month, or cyclical periods)

Specific performance requirements, related to a specific functional requirement, should be listed with that functional requirement.

#### **5.4.7 Capacity**

Not applicable to Project Hygge. [List the required capacities and expected volumes of data in business terms. Do not state capacities in terms of system memory requirements or disk space—if growth trends or projections are available, provide them]

#### **5.4.8 Data Retention**

[Describe the length of time various forms of data must be retained and the requirements for its destruction.

For example, “The system shall retain application information for 3 years”. Different forms of data include: system documentation, audit records, database records, access records.]

#### **5.4.9 Error Handling**

[Describe system error handling.]

#### **5.4.10 Validation Rules**

[Describe System Validation Rules.]

#### **5.4.11 Conventions/Standards**

[Describe system conventions and standards followed.

For example: Microsoft standards are followed for windows, Institute of Electrical and Electronics Engineers (IEEE) for data formats, etc.]

# **APPENDIX A - GLOSSARY**

[Define terms, acronyms, and abbreviations used in the FRD.]