**PRJ566 – Winter 2019**

**PRJ566 – Group No: 05**

**Name of Project: “MindSpark”**

**Project Leader: YoungMin Ko**

**Last updated: Mar 26, 2019**

**­­­­**

**Group Members:**

1. **Youngmin Ko**
2. **Gia Tuong Tran**
3. **Pratik Panchani**
4. **Arnav Bansal**

# Table of Contents

[Table of Contents 2](#_Toc4973559)

[1 – Introduction/Overview – Document Information 5](#_Toc4973560)

[1.01 – Document Authors 5](#_Toc4973561)

[1.02 – Revision History 6](#_Toc4973562)

[1.03 – Document Conventions 8](#_Toc4973563)

[1.04 – Document Purpose 9](#_Toc4973564)

[1.05 – Intended Audience 10](#_Toc4973565)

[1.06 – Acronyms & References 11](#_Toc4973566)

[1.07 – Group Agreement 12](#_Toc4973567)

[2 – Project Overview 14](#_Toc4973568)

[2.01 – Project Proposal 14](#_Toc4973569)

[2.02 – Project detailed scope & functionality 17](#_Toc4973570)

[2.03 – Stakeholders and Users 21](#_Toc4973571)

[2.04 – Business Opportunity 22](#_Toc4973572)

[2.05 – Risks 23](#_Toc4973573)

[2.06 – Constraints 24](#_Toc4973574)

[2.07 – Operating Environment 25](#_Toc4973575)

[2.08 – Operational, Performance & Security Requirements 26](#_Toc4973576)

[3 – Implementation Schedule for PRJ566 27](#_Toc4973577)

[3.01 – Work Breakdown Structure – PRJ566 – Design 27](#_Toc4973578)

[4. – Process/Activity & Data Modeling 28](#_Toc4973579)

[4.01 – Data Flow Diagrams 28](#_Toc4973580)

[User Sign-up 29](#_Toc4973581)

[User Sign-In 30](#_Toc4973582)

[User Request Password Reset 31](#_Toc4973583)

[User Password Reset 32](#_Toc4973584)

[Upload Quote 33](#_Toc4973585)

[Read Quote 34](#_Toc4973586)

[Delete Quote 35](#_Toc4973587)

[Rate Quote 36](#_Toc4973588)

[Report Quote 37](#_Toc4973589)

[Approve and Reject Quote 38](#_Toc4973590)

[Edit Profile (Admin profile manage) 39](#_Toc4973591)

[Machine Filter Management 40](#_Toc4973592)

[Edit Profile (User profile manage) 41](#_Toc4973593)

[Customize Reminder Setting 42](#_Toc4973594)

[5. – Project/System Design 43](#_Toc4973595)

[5.01 – Business Rules 43](#_Toc4973596)

[5.02 – Use Case Diagram 45](#_Toc4973597)

[5.03 – Use Case Specification 46](#_Toc4973598)

[UC01: Sign-up 46](#_Toc4973599)

[UC02: Sign in 48](#_Toc4973600)

[UC03: Sign Out 50](#_Toc4973601)

[UC04: Forgot Password 51](#_Toc4973602)

[UC05: Upload Quote 53](#_Toc4973603)

[UC06: Read Quote 55](#_Toc4973604)

[UC07: Delete Quote 57](#_Toc4973605)

[UC08: Rate Quote 59](#_Toc4973606)

[UC09: Report Quote 61](#_Toc4973607)

[UC10: Approve/Reject Quote 63](#_Toc4973608)

[UC11: Machine Filter Management 65](#_Toc4973609)

[UC12: Admin Profile Edit 67](#_Toc4973610)

[UC14: User Profile Edit 69](#_Toc4973611)

[UC15: Customize Reminder Settings 71](#_Toc4973612)

[5.04 Domain Class Diagram 73](#_Toc4973613)

[6. – Interface Mock-ups 74](#_Toc4973614)

[6.01 - Main page 74](#_Toc4973615)

[6.02 – Sign up 75](#_Toc4973616)

[6.03 – Sign in 76](#_Toc4973617)

[6.04 – Forgot Password 77](#_Toc4973618)

[6.05 – Upload quote 78](#_Toc4973619)

[6.06 – Reading quote 79](#_Toc4973620)

[6.07 – Delete quote 80](#_Toc4973621)

[6.08 – Rate quote 81](#_Toc4973622)

[6.09 – Report quote 82](#_Toc4973623)

[6.10 – Admin profile manage/approve 83](#_Toc4973624)

[6.11 – Admin profile manage/reject 84](#_Toc4973625)

[6.12 – Admin profile manage/edit 85](#_Toc4973626)

[6.13 – Admin profile manage/machine filter management 86](#_Toc4973627)

[6.14 – User profile manage / profile edit 87](#_Toc4973628)

[6.15 - User profile manage / customize reminder setting 88](#_Toc4973629)

[7.0 – Database 89](#_Toc4973630)

[7.01 – Scripts to Create Database Tables 89](#_Toc4973631)

[7.02 – Scripts to Insert Data(meaningful data) 93](#_Toc4973632)

[7.03 – Scripts to Delete Tables: 96](#_Toc4973633)

[7.04 – Scripts to populate tables (Alternate SQL queries) 97](#_Toc4973634)

[7.05 – Data Dictionary 99](#_Toc4973635)

[8. – Implementation Schedule for PRJ666 (week 12) 102](#_Toc4973636)

[8.01 – Schedule (Make sure to include names, dependencies, start and finish dates and hours) 102](#_Toc4973637)

[9 – Measurable deliverables 103](#_Toc4973638)

[10 – Acceptance Criteria 104](#_Toc4973639)

[11 – Client / Faculty Sign-off 105](#_Toc4973640)

[12 - PRJ666 Virtual Machine (VM) Requirements 106](#_Toc4973641)

# 1 – Introduction/Overview – Document Information

## 1.01 – Document Authors

1. YoungMin Ko
2. Gia Tuong Tran
3. Pratik Panchani
4. Arnav Bansal

## 1.02 – Revision History

|  |  |
| --- | --- |
| Week 04 | Started and completed Project Initiation sections:   * [1.01](#_1.01_–_Document) – Document Authors * [1.02](#_1.02_–_Revision) – Revision History (ongoing) * [1.03](#_1.03_–_Document) – Document Conventions * [1.04](#_1.04_–_Document) – Document Purpose * [1.05](#_1.05_–_Intended) – Intended Audience * [1.07](#_1.07_–_Group) – Group Agreement * [2.01](#_2.01_–_Project) – Project Proposal * [2.02](#_2.02_–_Project_1) – Project detailed Scope & functionality |
| Week 05 | Completed the following sections:   * [2.03](#_2.03_–_Stakeholders_1) – Stakeholders and Users * [2.05](#_2.05_–_Risks_1) – Risks * [2.06](#_2.06_–_Constraints_1) - Constraints * [2.07](#_2.07_–_Operating_1) – Operating Environment * [2.08](#_2.08_–_Operational,) – Operational, Performance and Security Requirements |
| Week 06 | Edited and improved the following:  [2.02](#_2.02_–_Project_1) – Project detailed Scope & functionality  Completed the following sections:  [4.1](#_4.01_–_Data) – Data Flow Diagrams  [5.1](#_5.01_–_Business_1) – Business Rules |
| Week 07 | Completed the following sections:   * [5.2](#_5.02_–_Use_1) - Use Case Diagram * [5.4](#_5.04_Domain_Class) - Domain Class Diagram   Started the following sections:   * [5.3](#_5.03_–_Use_1) - Use Case Specifications * [6.](#_6._–_Interface) - Interface Mock-ups |
| Week 08 | Worked on the following section:   * [5.3](#_5.03_–_Use_1) - Use Case Specifications * [6.](#_6._–_Interface) - Interface Mock-ups |
| Week 09 |  |
| Week 10 | Worked on the following section:   * [5.3](#_5.03_–_Use_1) - Use Case Specifications * [6.](#_6._–_Interface) - Interface Mock-ups |
| Week 11 | Completed the following sections:   * [5.3](#_5.03_–_Use_1) - Use Case Specifications * [6.](#_6._–_Interface) - Interface Mock-ups * [7.1](#_7.01_–_Scripts) - Scripts to create tables * [7.2](#_7.02_–_Scripts) - Scripts to populate tables (meaningful data) * [7.3](#_7.03_–_Scripts) - Scripts to delete tables * [7.4](#_7.04_–_Scripts) - Data Dictionary |
| Week 12 | Completed the following sections:   * [8.](#_8._–_Implementation) - Creating a work breakdown structure PRJ666 - Implementation part * [9.](#_9_–_Measurable_1) - Deliverable list for PRJ666 * [10.](#_10_–_Acceptance_1) - Acceptance Criteria * [11.](#_11_–_Client_1) - Client / Faculty Sign-off * [12.](#_12_-_PRJ666) - PRJ666 - VM Requirements |
| Week 13 |  |
| Final |  |

## 1.03 – Document Conventions

Any text highlighted in red indicates an exception or error

Anything highlighted in blue is in-progress

Any text highlighted in yellow is an important point.

Any text *italicized* represents definitions.

## 1.04 – Document Purpose

This document provides the informational documentation of specifications, features and functionalities related to “mindSpark” Motivational Web Application. It acts as a detail manual of tools and services involved in building this application. This document also reflects detail descriptions of various stockholders, references and resources involved.

## 1.05 – Intended Audience

The intended audience for this application would be those who wants to be motivated, read, rate and share quotes based on their mind state or mood. This application is also targeted to those people who wants to upload their own quotes and ideas. Moreover, public admirers can customize email notifications based upon their preferences.

## 1.06 – Acronyms & References

## 1.07 – Group Agreement

**Group:**

prj566\_191a05(group #5)

**Project Title:**

“mindSpark”

**Project Time Frame:**

27/01/2019 – 17/08/2019

**Group Members:**

Gia Tuong Tran

Pratik Panchani

Youngmin Ko

Arnav Bansal

**Group Leadership:**

YoungMin Ko

**Team Functions:**

* *We will share the information through Discord, Email and Google Drive*
* **Youngmin Ko**: Group Leader, take charge of presentation, providing fundamental ideas of app. Manage team work environment.
* **Pratik Panchani**: In charge for content collection, research and database for the application.
* **Gia Tuong Tran**: In charge of creating website and programming
* **Arnav Bansal**: Developer for the main application system and AI.

**Group Meetings:**

* Every Thursday at 5pm in Library

**Group Problems:**

* We have various emergency contact network and installed discord app on each team member’s phone. We verified that we can communicate in real time. We encourage each team members to not fall behind. We will assign team members on each one’s best field.

**Team Commitment**

**The undersigned members agree to work together on the project until the end of the PRJ666 next Semester. They recognize that as a team and individually they are responsible for the quality of all deliverables.**

**Name Date**

**Gia Tuong Tran 27/01/2019**

**Youngmin Ko 27/01/2019**

**Pratik Panchani 27/01/2019**

**Arnav Bansal 27/01/2019**

# 2 – Project Overview

## 2.01 – Project Proposal

Team #: 05

Team Members:

* Gia Tuong Tran
* Pratik Panchani
* Young Min Ko
* Arnav Bansal

**Brief description of system you are proposing. Please include your research reference (contacts, etc.):**

*“mindSpark” is an app that encourages people through quotes and messages. It helps people who are having tough time with emotional problem. It also lets you upload your own quotes which can be shared to help other people.*

**Please explain why you feel there is a need or market for this system.**

Mental illness is very common in modern society. It’s far more common than we know. In times past, the technology evolved without research with humanities. However, now many people acknowledge that technology should be evolved with humanities. Now people know how important humanity is and how to use technology to improve humanity. We, as developers will use technology(program) to care humanity (people’s mind). The app will provide virtual space that people take care of each other.

**Please describe the system in more detail. Choose the functions that you feel are most important and describe how they will work. Give each function a title and briefly explain its responsibility in a few sentences.**

Every user will have their own account. They can set their state through setting like “happiness”, “depression”, “satisfactory” and “normal” and from that “mindSpark” will daily send them the quote. Also, they can upload their own quote and share it with everybody else.

Functionalities:

Receiving periodic quote and sending feedback to update user’s emotional state

After setting up user’s current mood, they can receive periodic quote message. When user receives quote, the app asks user feedback: how helpful is this quote, is state of mind changed or report as an inappropriate quote. The user feedback will be applied to the ‘quote sending logic’ which means, the user feedback will the change of the result of what user will receive in future.

Upload quote

Users can upload their own quote to “mindSpark”. The quote will be filtered by the program (if the quote contains some expletive) and confirmation by the administrator to be able to upload.

Quote Rating

When the user uploads a quote. Everyone can rate that quote from the scale from 1 to 5 star. If the quote has good rate, it will be recommended to people more often. If the user has many good rated quotes, he/she will receive an award. If it receives bad reviews or reported, it might be deleted.

**Please walk through a typical usage of your system (a scenario). For example, selecting and selling an item; creating an appointment, setting up a sports team.**

Create an account:

When the users use the application, “mindSpark” will require users to create an account with name, age, email and current emotion. The application will show some default choices for the emotion like “normal”, “weariness” and “happiness”. Users can choose from that or they can type in what mood are they in now. After that, users can set up reminder schedule. The default option is 8 am everyday but it can be customized (including turn off).

Receiving quote:

After the basic setup, now user can receive quote reminder. When user receives quote, the app asks user feedback: how helpful is this quote, is state of mind changed or report as an inappropriate quote. The user feedback will be applied to the ‘quote sending logic’ which means, the user feedback will the change of the result of what user will receive in future.

Updating user emotion:

The app asks user emotion and state of mind daily. However, app will never ask user a subjective question. The user will not ‘type’ to answer but do some simple gesture such as click or slide.

Upload quote:

You not only receive quote daily, but you can also write your own quote and share it with everyone. You can view other people’s quote and rate it too. The uploaded quote is filtered by program and confirm by administrator. Once it’s confirmed, the quote will be upload publicly and people can rate it. If it receive bad reviews or reports, the quote might be removed

## 2.02 – Project detailed scope & functionality

We will develop a web app called “mindSpark” that encourage people who are having tough time with emotional problem. It also lets users upload their own quotes which can be shared to help other people. The main purpose is support people’s healthy state of mind so help people to get over emotional issue.

In our application, we will have the functions and features listed below:

**Main Page**

* Display user’s favorite quotes, popular and high-rated quotes that related to the user state setting
* Users can rate and report other user uploaded quotes
* Link to sign up and sign in
* Link to personal profile
* Link to account setting
* Option to choose another mood
* Developer contact at the bottom

Form elements:

* User generated quote upload (textbox) – the user can input alphanumeric characters (Minimum 10 characters)
* Rating button – Users can choose from 1 to 5 star depend on how good the quote is. If the quote is received more than 20 one start, then the quote will be transferred to the pending line of author accounts to check it should be removed or stayed
* Report button – Users can report the quote if you think that quote is bad or contains some expletive. The reported quote will be transferred to the pending line of author account to review it.
* Sign-up button – (Users not sign in yet) Link users to the sign-up page if they want to register new account
* Sign-in button – (Users not sign in yet) Link users to the sign-in page
* Setting button – Link users to account setting button if they want to change or update personal information
* Choose emotional state – user can change their mood in order to see other quotes or they can choose All to see all the default quotes and the most high-rated quotes.

**Sign up page**

* User sets up own account (it’s mandatory). Put name, age, email, and current emotion and state of mind.
* Before setting up the state of mind, the app shows the user what kinds of option do they have such as ‘normal’, ‘weariness’, ‘depression’, ‘tedious’, ‘panic disorder’, ‘wimpishness’, ‘happiness’, ‘satisfactory’ and etc.
* User will pick one or more state of mind. After that, user will set up the reminder schedule. The default option is once a day 8 a.m. morning but it can be customized (including turn off option).
* User email should be verified to sign up.

Form elements:

* Username (textbox) – the user can input alphanumeric characters. (Maximum 25 characters)
* Password (textbox) – the user can input alphanumeric characters. (Maximum 25 characters)
* Email (textbox) – the user can input alphanumeric characters and check if it is a valid email format
* Date of birth (select list) – the user can select their date of birth
* State of mind (select list) – the user can select from a list of states of mind. (Maximum 7 item)
* Preferred alert time (select list) – the user can select preferred quote alert time. (Minimum 0 and Maximum 5)
* Sign-up button – User clicks this to submit sign up form. The system will check all the required fields are filled out, query the database to see if the username already exists and will display an error message to try other username. Otherwise, it will submit the request and display a welcome message.

**User Profile Page**

The profile page will include all the personal information and the list of uploaded quotes of currently logged user. This page also contains settings for quote alert preference. Here user can manage and edit this information.

* Left Panel: User account information which includes currently logged user information as given below:
  1. User Name (Text Field) - Unique for each account
  2. First Name (Text Field)
  3. Last Name (Text Field)
  4. Email Address (Text Field) - Single email can be registered only once
  5. Password (Password Text Field)
  6. Date of Birth (Text Field)

**NOTE: All of the above fields can be edited, and all the validation rules apply similar to the sign-up page.**

1. Save (Button) – To save the edited information

* Middle Panel: Currently logged user uploaded quotes with the following statistics:

1. Average Rating – It is in the form of filled star
2. Edit (Button) – To edit the uploaded quote
3. Status (Label) – Under review or Approved
4. Delete (Button) – To delete the quote

* Right Panel: Preference for quote alert via email with following settings which can be edited:

1. Current Mind State/mood (Label) – Currently select Mind State
2. List of available Mind state/mood (List) – Click on the available categories of mindState/mood to reset it.
3. Reminder alert with time selection options (Timer) – Select the specific time to send the email with quote
4. Days (7 Checkbox) – Check for which days to send the email from Monday to Sunday
5. Set (Button) – To set the reminder

**Admin Profile Page**

The admin profile page is similar to the normal user profile page, but it allows more privileges for other features. This feature includes the right to approve and reject the under-review quotes.

* Left Panel: Admin account information which includes currently logged user information as given bellow:

1. User Name (Text Field) - Unique for each account
2. First Name (Text Field)
3. Last Name (Text Field)
4. Position (Text Field)
5. Email Address (Text Field) - Single email can be registered only once
6. Password (Password Text Field)
7. Date of Birth (Text Field)

**NOTE: All of the above fields can be edited, and all the validation rules apply similar to the sign-up page**

1. Save (Button) – To save the edited information

* Middle Panel: List of pending quotes to be reviewed with the link to uploader's profile page

1. Message field - To be sent to the uploader after the decision.
2. Approve
3. Reject

* Right Panel: Preference for quote alert via email with following settings which can be edited:

1. Current Mind State/mood (Label) – Currently select Mind State
2. List of available Mind state/mood (List) – Click on the available categories of Mind state/mood to reset it.
3. Reminder alert with time selection options (Timer) – Select the specific time to send the email with quote
4. Days (7 Checkbox) – Check for which days to send the email from Monday to Sunday
5. Set (Button) – To set the reminder

**Forgot Password Page**

This page is for the users/members in case they have forgotten the password for their account. Members can use this page to request for their password to be changed. After the submission of this page, the system will send a link to the email provided in the form if a corresponding account is active for that email address.

Form elements:

* This page will have a textbox where the user can input either their username, or their email address and a submit button to submit the form to get a password reset link on their email.
* *Username/Email textbox* – the user can input alphabetical characters or numbers. There will be a constraint on the number of characters (maximum 30 characters) unless there is a ‘@’ character, indicating that the user is inputting an email and not a username. If there is a ‘@’ in the *textbox,* it will be checked for a valid email format.
* *Submit button* – on clicking of this button, the database will be checked to see if a matching account exists for the provided email or username. If an account exists, an email with the password reset link for the provided account will be sent to the email otherwise, an error message will be displayed indicating that the account with the provided email or username does not exist.

## 2.03 – Stakeholders and Users

The main goal of “mindSpark” is to encourage people to overcome own emotional issue through quotes and messages. The stakeholders would consist of:

* Development Team
* Administrator
* Investors/Sponsors
* Individual that facing emotional problem
* Organization that manages or treats patients who are having mental issues
* organization that needs to motivate employees
* Organizations and Individuals who are looking for people support and help building habits

## 2.04 – Business Opportunity

Unfortunately, mental illness is very common in the modern society. It’s like a cold or flu in human mind. It’s far more common than we know. In old days, the technology evolved without research with humanities. However, now many people acknowledge that technology should be evolved with humanities. Now people know how important humanity is and how to use technology to improve humanity. People actively look for apps to improve their life quality with the health-related app and they will find interest in “mindSpark” which will take care of their emotional health.

## 2.05 – Risks

|  |  |
| --- | --- |
| Risk | Response |
| Not be able to complete the task on time | * Other members can help to finish the task, group leader should notify other members and professor so it can be rescheduled |
| Members absent from group meeting | * Member should have Discord or video call for the meeting or reschedule another meeting |
| Difficult for the users to continue writing quotes and upload onto application | * Target specifically writers and bloggers to upload the quotes |
| Supporting mobile platform | * Make the website responsive. Therefore, easy to use in mobile web browser |
| Unfair participation of team work causes ruining team spirit | * Group will consider re-assign each task |

## 2.06 – Constraints

* Because the time and man power are limited, our app “mindSpark” will be available in English only. The critical reason is filtering and translating user-generated-quote.
* The users will need to have internet connection to use the app because it’s web based.
* If the quote is sent to the review list. It will take minimum 24 hours to receive the decision and upload it to the public list

## 2.07 – Operating Environment

The application will be a web application which will be made accessible on all platforms, Windows, Linux and MacOS, using a web browser.

The application will be built using Visual Studio Code IDE. The IDE can be used on Windows, Linux or MacOS systems. The application will be built using several technologies including JavaScript and NodeJS for the web application server, MongoDB for the application database. The application build environment will be maintained by using GIT to sync and update the code base for the application.

The application will be tested on main-stream web browsers, Google Chrome, Mozilla Firefox and Safari on all three major OS, Windows, Linux and MacOS.

## 2.08 – Operational, Performance & Security Requirements

The application will be able on most of OS which has modern web browser. Most of the features will require internet connection especially receiving and sending new quotes.

Passwords will be enforced with a strict password policy and stored as a hashed string in the database. If user forgets password, they will have to answer some security question. If the information is correct, they will get an email with an expiring link to reset password.

It won’t take longer than one or two minutes to upload the quote unless the user-generated-quote fails to pass through the filter. If the quote successfully passes the filter, it will be transferred to the pending line in the admin account to check and the admin will examine the context to find if there is any hidden message. This will take up to within 24 hours for the admin to check the quote.

# 3 – Implementation Schedule for PRJ566

## 3.01 – Work Breakdown Structure – PRJ566 – Design

<https://zenit.senecac.on.ca/~prj566_191a05/group5_msproject.mpp>

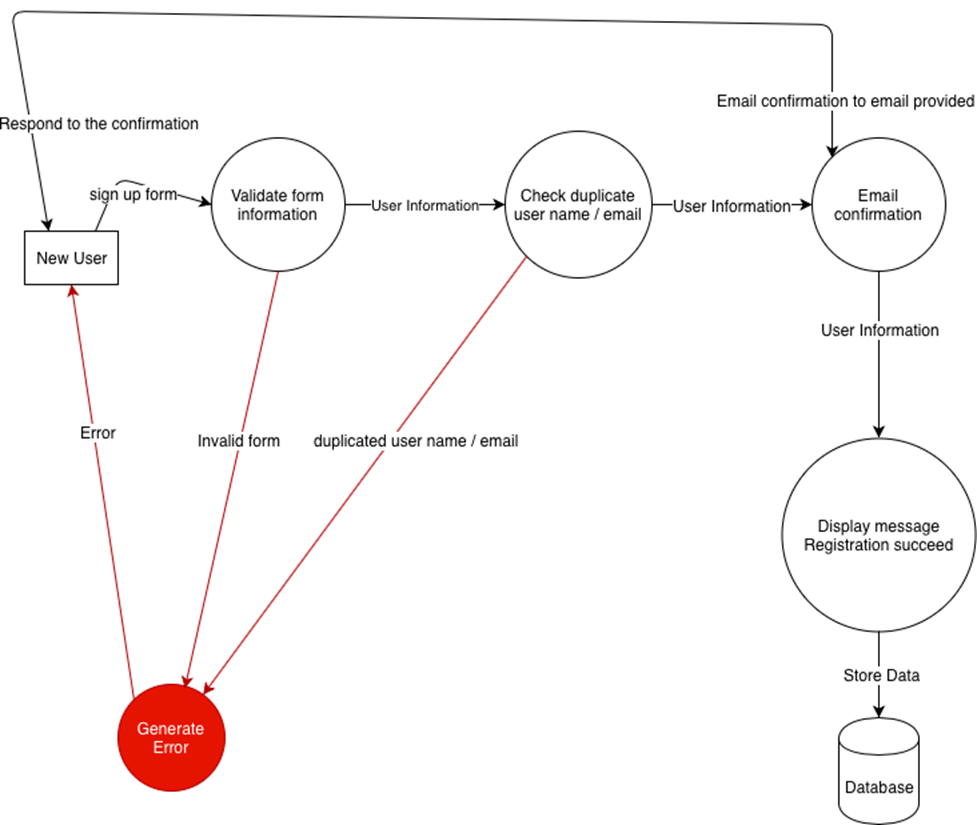
# 4. – Process/Activity & Data Modeling

## 4.01 – Data Flow Diagrams

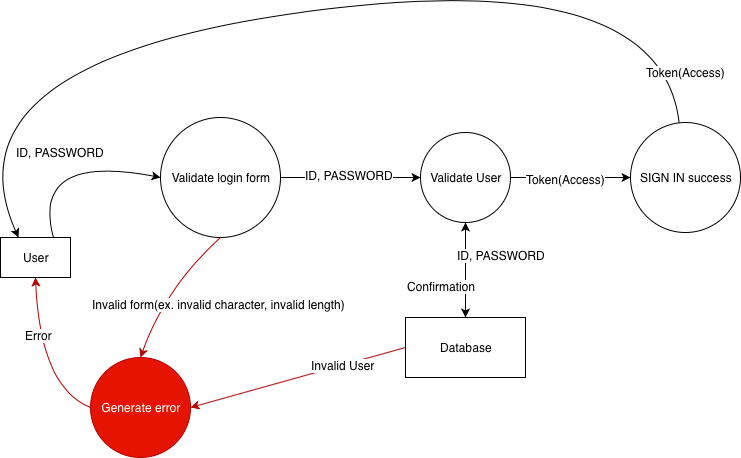
Data flow Diagram Lists

* About access
  + Sign up
  + Sign in
  + Sign forget password
* About manage Quote
  + Upload quote
  + Read quote
  + Delete quote
  + Rate quote
  + Report quote
* About admin profile manage
  + Approve
  + Reject
  + Profile Edit (Admin profile manage)
  + Machine filter management
* About user profile manage
  + Profile Edit (User profile manage)
  + Customize reminder setting

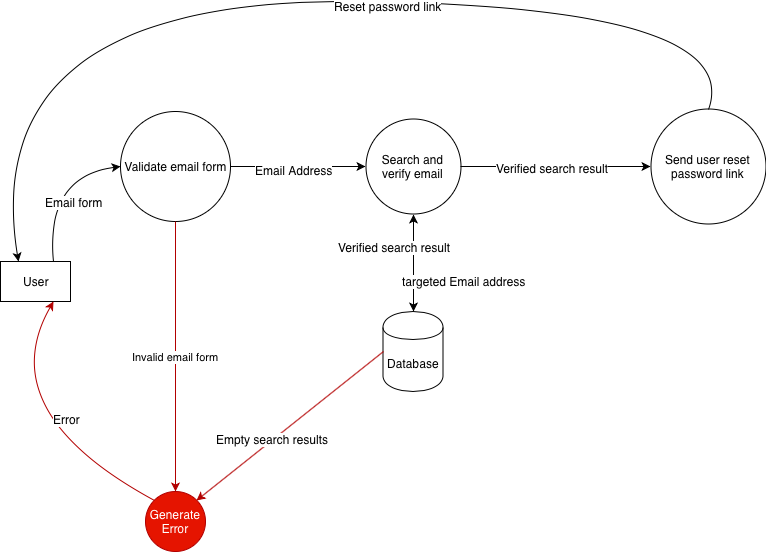
### User Sign-up



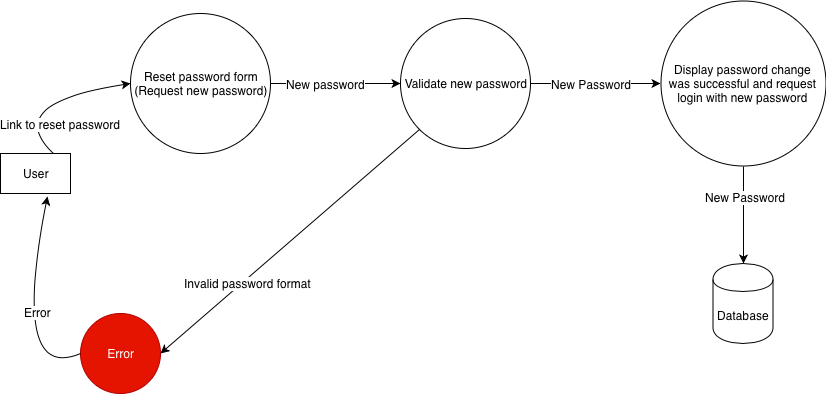
### User Sign-In

****

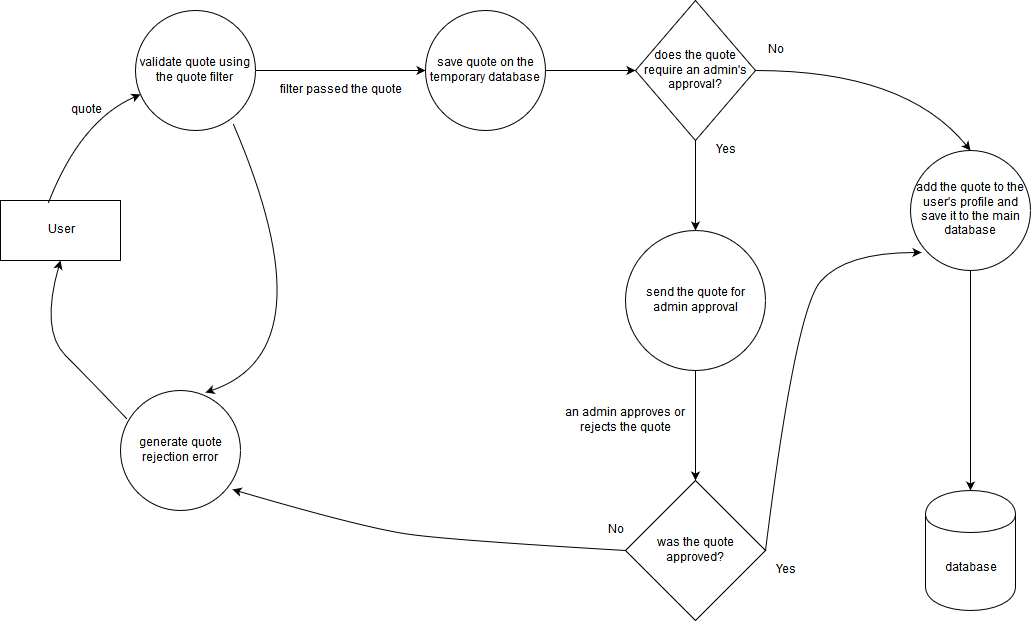
### User Request Password Reset

****

### User Password Reset

****

### Upload Quote



### Read Quote



### Delete Quote



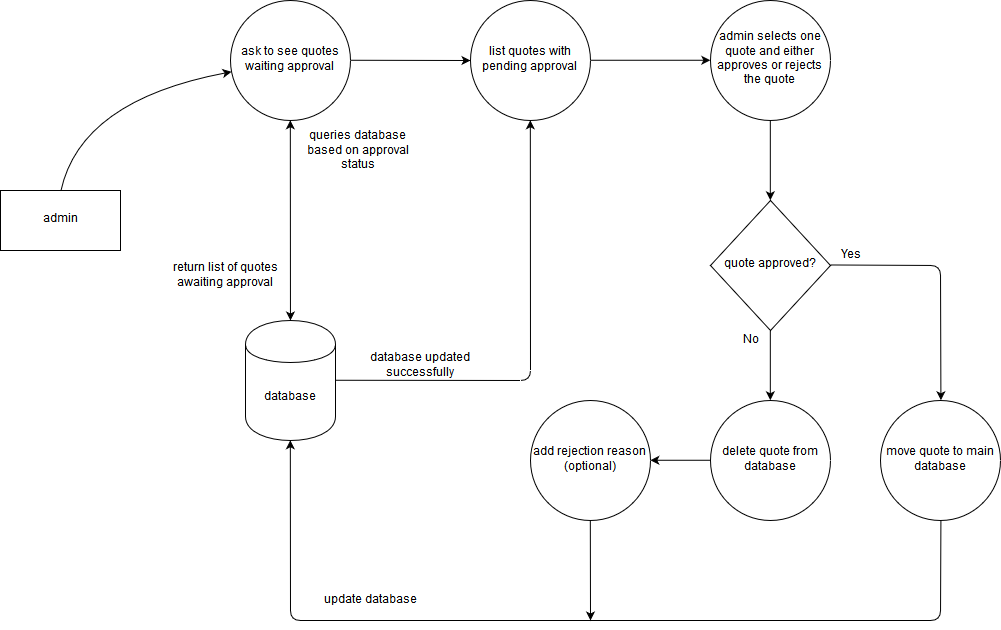
### Rate Quote



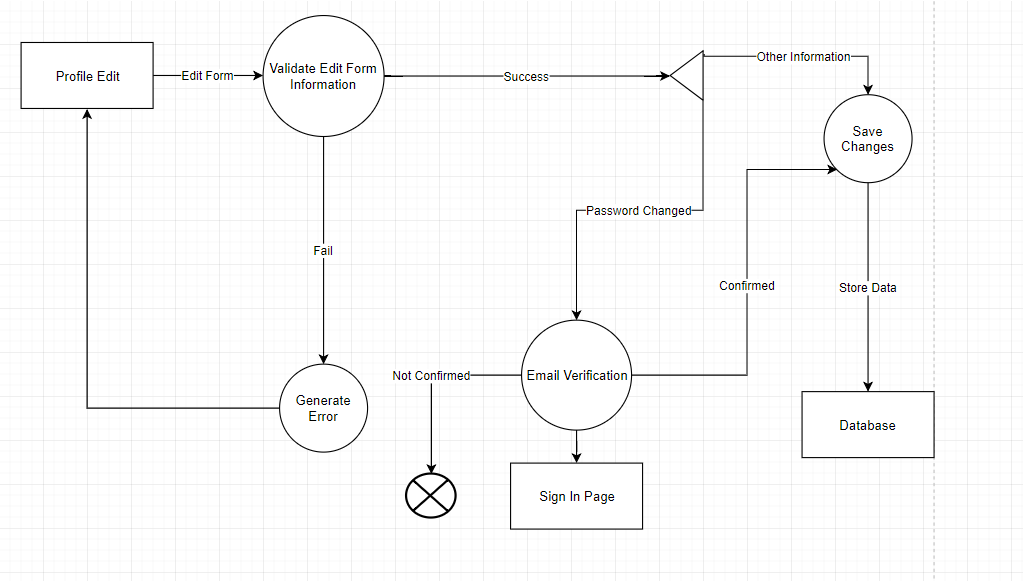
### Report Quote



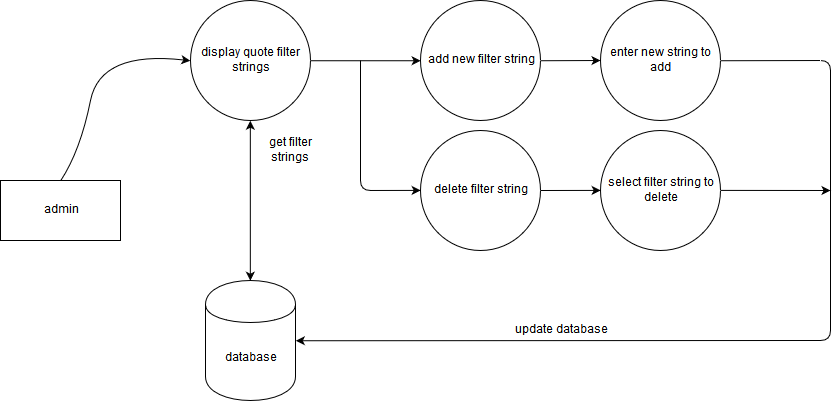
### Approve and Reject Quote



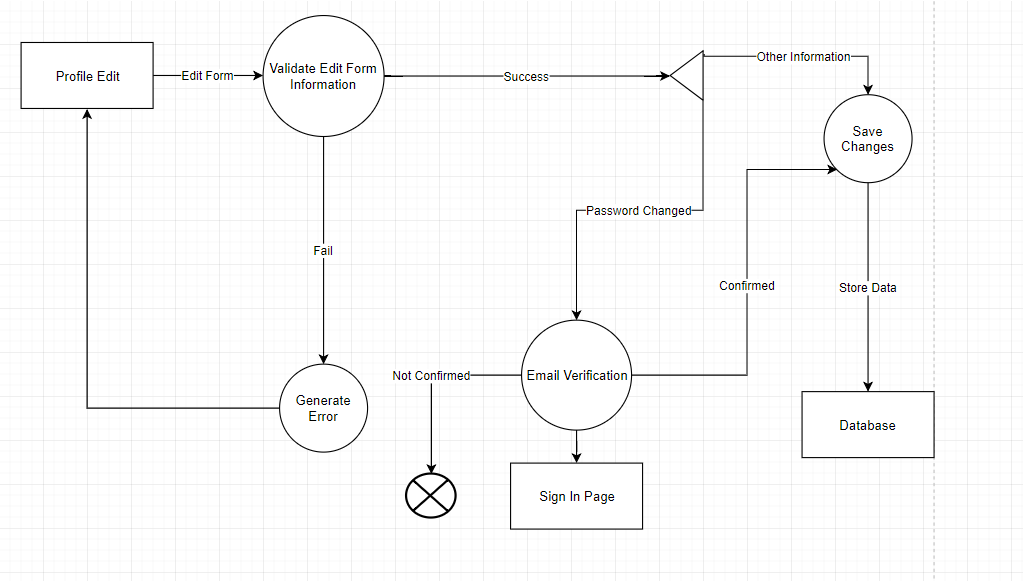
### Edit Profile (Admin profile manage)



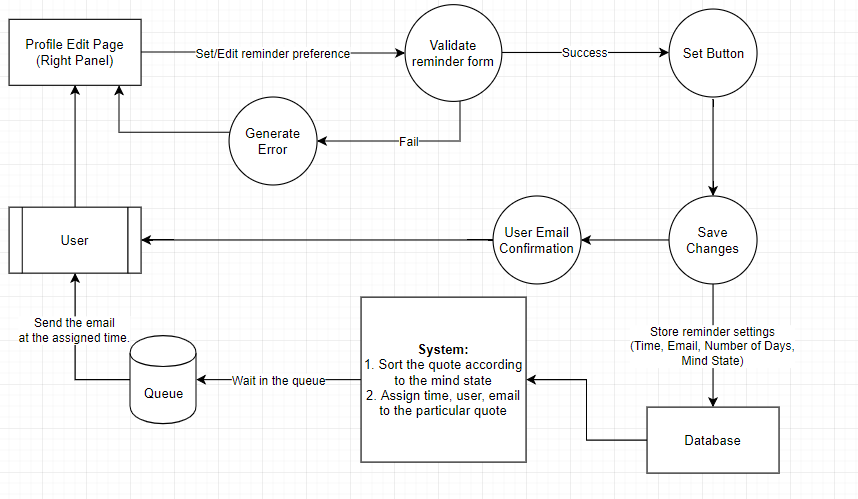
### Machine Filter Management



### Edit Profile (User profile manage)



### Customize Reminder Setting



# 5. – Project/System Design

## 5.01 – Business Rules

| Business Rule Number | Business Rule Description | Related UC |
| --- | --- | --- |
| BR01 | The Username can contain alphanumeric characters and no more than 25 characters. | UC01, UC12, UC14 | |
| BR02 | The password can contain alphanumeric characters and no more than 25 characters. | UC01, UC04, UC12, UC14 | |
| BR03 | Date of birth will be formatted YYYY-MM-DD. | UC01, UC12, UC14 | |
| BR04 | Email must be a valid email address format and verified. | UC01, UC12, UC14 | |
| BR05 | Maximum 7 State of mind can be selected at once. | UC14 | |
| BR06 | Minimum 0, Maximum 5 Preferred alert time can be selected. | UC15 | |
| BR07 | Password can be changed only after email verification. | UC04, UC12, UC14 | |
| BR08 | New username should be unique | UC01, UC12, UC14 | |
| BR09 | Quote report reason cannot contain special characters or expletives | UC09 | |
| BR10 | One user can only select one mind state at one time for reminder | UC15 | |
| BR11 | One user can only have one email to set the quote reminder | UC15 | |
| BR12 | User can only set one quote reminder per day on selected week days which repeats every week. | UC15 | |
| BR13 | One user can report one quote at one time | UC09 | |
| BR14 | one user can delete one personal quote at one time |  | |
| BR15 | one user can rate one quote at the time | UC08 | |
| BR16 | One author can check many quotes at one time | UC06 | |
| BR17 | If a filter string needs to be changed for the quote filter, the admin will first have to delete the existing string that needs to be changed and then create a new one with the updated string | UC11 | |
| BR18 | If there are 2 or more matches in the newly uploaded quote with the filter strings, the quote will be rejected automatically by the quote filter | UC11 | |
| BR19 | No matches with the filter strings in a newly uploaded code will lead to an automatic approval | UC11 | |
| BR20 | One user can upload one quote at one time | UC05 | |
| BR21 | Quote cannot contain expletive | UC11 | |
| BR22 | User can view multiple quotes at the time | UC06 | |

## 5.02 – Use Case Diagram

A picture containing text, map

Description automatically generated

## 5.03 – Use Case Specification

### UC01: Sign-up

**Identifier**: UC01

**Brief description**: This use case enables people who want to become a member of “mindSpark” to submit their sign-up forms.

**Section 1: Business Rule(s):**

* BR01
* BR02
* BR03
* BR04
* BR05

**Section 2**: Main Scenario with Alternative Flows:

**Main Flow**: Submitting sign-up form to become a member

**Preconditions**:

* The user is on the Sign-up page.
* The system is running properly.

|  |  |  |  |
| --- | --- | --- | --- |
| **Step #** | **Actor: User** | **System** | **Data Used** |
| 1 | Clicks “sign-up” button at the main page | Display the sign-up page |  |
| 2 | Fill out the sign-up form | Validate all user inputs. Checking the format of user input | Username, password, email, dateOfBirth, StateOfMind, PreferredAlertTime |
| 3 | Receive verification email and response to verify | Send email verification to user email and receive user response | Email, emailVerified |
| 4 | Submit sign-up form | Save the request to database | Username, password, email, emailVerified , dateOfBirth, StateOfMind, PreferredAlertTime |

**Alternative Flow**

|  |  |  |
| --- | --- | --- |
|  | Alternative Flow | Description |
| 1 | Incorrect input format | The system displays error message, high light the incorrect part of form and let user to fill out the form again |

**Section 3: Class/Attributes identified:**

|  |  |
| --- | --- |
| Class | Attribute of Class |
| Member | Username, password, email, emailVerified , dateOfBirth, StateOfMind, PreferredAlertTime |

**Section 4: Exceptions Identified:**

|  |  |
| --- | --- |
| **Exception** | **How it will be handled** |
| User Email Verification Failure | Display error message and alert that the system will send user another verification email |
| Form Submission Failure | Display error message |

### UC02: Sign in

**Identifier**: UC02

**Use Case Specification for:** Sign in

**Brief Description:** This use case enables user and admin to sign into the application.

**Section 2: Main Scenario with Alternative Flows (HD):**

**Main Flow:**

**Preconditions:**

The user or admin is on the main page and the system is ready

| **Step #** | **Actor: Employee** | **System** | **Data Used** |
| --- | --- | --- | --- |
| 1 | Clicks the “Sign in” button | Prompts for username and password |  |
| 2 | Enters the username and password, clicks on the “Sign in” button | Validates the username and password. Log in the application and start session. | Username, Password, Position |

**Alternative Flow**

|  | **Alternative Flow** | **Description** |
| --- | --- | --- |
| 1 | The validation fails | System displays an error message saying that credentials are incorrect |

**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| **Class** | **Attribute of class** |
| User | Username, password, position |

**Section 4: Exceptions identified:**

|  |  |
| --- | --- |
| **Exception** | **How it will be handled** |
| Too many login attempts | Disable the account and ask user to reset password thorough forgot password OR verify through email |
| Server error | Display user friendly error page |

### UC03: Sign Out

**Identifier**: UC03

**Use Case Specification for:** Sign out

**Brief Description:** This use case enables user and admin to sign out of the application.

**Section 2: Main Scenario with Alternative Flows (HD):**

**Main Flow:**

**Preconditions:**

The user is signed on.

| **Step #** | **Actor: Employee** | **System** | **Data Used** |
| --- | --- | --- | --- |
| 1 | Clicks the “Sign out” button at the top right of the display | Ask user for confirmation |  |
| 2 | Clicks the “Confirm” button | Sign out the current user and end the session |  |

**Alternative Flow**

|  | **Alternative Flow** | **Description** |
| --- | --- | --- |
| 1 | Clicks the “Cancel” button | Keep the user signed in. |

**Section 4: Exceptions identified:**

|  |  |
| --- | --- |
| **Exception** | **How it will be handled** |
| Server error | Display user friendly error page |

### UC04: Forgot Password

**Identifier**: UC04

**Brief Description:**

This use case enables the user or admin to reset the password

**Section 1: Business Rule(s):**

* BR02- The password can contain alphanumeric characters and no more than 25 characters.
* BR04- Email must be a valid email address format and verified.
* BR07- Password can be changed only after email verification.

**Section 2: Main Scenario with Alternative Flows (HD):**

**Main Flow:**

**Preconditions:**

The user and admin are on the main page and not signed in.

| **Step #** | **Actor: Agent/Admin** | **System** | **Data Used** |
| --- | --- | --- | --- |
| 1 | Click “Forgot Password” button | Display the Forgot password form |  |
| 2 | User types registered email address and clicks on “Send verification email” | Validate the email and send the verification email to the user’s email. Display message to access the email and reset the password | Email address |
| 3 | Click the reset link on the email | Validate the link and display the validation form | New password |
| 4 | Types in the new password, confirm new password, and presses submit | Validate the password and check the rules on the new password. If yes, updates the database with the new password | password |

**Alternative Flow**

|  | **Alternative Flow** | **Description** |
| --- | --- | --- |
| 1 | No registered email address found | No user found in database. Ask user to put in correct email |
| 4 | System fails to validate the link and displays an error message | The validation link in incorrect |
| 5 | System fails to validate the rules of password according to rules | Display the error message to input correct password |

**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| **Class** | **Attribute of class** |
| User | Email address, password |

**Section 4: Exceptions identified:**

|  |  |
| --- | --- |
| **Exception** | **How it will be handled** |
| Server error | Display user friendly error page |

### UC05: Upload Quote

**Identifier**: UC05

**Brief Description**: This use case enables the user to upload quote on “mindSpark”. It might take up to 5 minutes to upload.

**Section 1**: Business Rule(s):

* BR20: User can upload one quote at one time
* BR09: Quote cannot contain expletives

**Section 2**: Main Scenario with Alternative Flows (HD):

**Main Flow**: Upload Quote

**Preconditions**:

The user is on the main page and already signed in.

| Step # | Actor: User | System | Data Used |
| --- | --- | --- | --- |
| 1 | Clicks on the text area on the main page |  |  |
| 2 | Type the quote |  |  |
| 3 | Clicks on Submit button | Displays a successful message | *Message, UploadedQuote* |

**Alternative Flow**

|  | Alternative Flow | Description |
| --- | --- | --- |
| 1 | Quote contain expletives | Display the warning message about the bad words on the quote and return to the main page |

**Section 3: Classes/Attributes identified**:

|  |  |
| --- | --- |
| Class | Attribute of class |
| Quote | UploadedQuote, Message |

**Section 4: Exceptions identified**:

|  |  |
| --- | --- |
| Exception | How it will be handled |
| Submission Failure | Display error message |
| Server error | Display a friendly error page |

### UC06: Read Quote

**Identifier**: UC06

**Brief Description**: This use case enables the user to view related-to-the-mood quotes.

**Section 1**: Business Rule(s):

* BR22: User can view multiple quotes at one time

**Section 2**: Main Scenario with Alternative Flows (HD):

**Main Flow**: Read Quote

**Preconditions**:

The user is on the main page and already signed in.

| Step # | Actor: User | System | Data Used |
| --- | --- | --- | --- |
| 1 | Clicks on the Mood Option on the left | Display list of related quotes that rank from five star to one star |  |

**Alternative Flow**

|  | Alternative Flow | Description |
| --- | --- | --- |
| 1 |  |  |

**Section 3: Classes/Attributes identified**:

|  |  |
| --- | --- |
| Class | Attribute of class |
|  |  |

**Section 4: Exceptions identified**:

|  |  |
| --- | --- |
| Exception | How it will be handled |
| Server error | Display a friendly error page |

### UC07: Delete Quote

**Identifier**: UC07

**Brief Description**: This use case enables the user to delete user’s quote.

**Section 1**: Business Rule(s):

* BR14: User can delete one quote at one time

**Section 2**: Main Scenario with Alternative Flows (HD):

**Main Flow**: Delete Quote

**Preconditions**:

The user is on the main page and already signed in.

| Step # | Actor: User | System | Data Used |
| --- | --- | --- | --- |
| 1 | Clicks on the Down Arrow Icon beside user’s quote |  |  |
| 2 | Clicks on Delete Quote option | Displays a pop-up window and asking for final confirmation |  |
| 3 | Clicks on Submit button | Displays a confirm message | *Message* |

**Alternative Flow**

|  | Alternative Flow | Description |
| --- | --- | --- |
| 1 |  |  |

­­­

**Section 3: Classes/Attributes identified**:

|  |  |
| --- | --- |
| Class | Attribute of class |
| Quote | Message |

**Section 4: Exceptions identified**:

|  |  |
| --- | --- |
| Exception | How it will be handled |
| Submission Failure | Display error message |
| Server error | Display a friendly error page |

### UC08: Rate Quote

**Identifier**: UC08

**Brief Description**: This use case enables the user to rate quote. It might take up to 24 hours to the administrator to check the quote.

**Section 1**: Business Rule(s):

* BR15: User can rate one quote at one time
* BR16: Administrator can check multiple quotes at one time

**Section 2**: Main Scenario with Alternative Flows (HD):

**Main Flow**: Rate Quote

**Preconditions**:

The user is on the main page and already signed in.

| Step # | Actor: User | System | Data Used |
| --- | --- | --- | --- |
| 1 | Clicks on the Down Arrow Icon beside the quote |  |  |
| 2 | Clicks on Rate Quote option | Displays a pop-up window and prompts to choose from one to five stars depend on how good it is and optional comment | *RatedQuote, Comment* |
| 3 | Clicks on Submit button | Displays a confirm message | *Message* |

**Alternative Flow**

|  | Alternative Flow | Description |
| --- | --- | --- |
| 1 |  |  |

**Section 3: Classes/Attributes identified**:

|  |  |
| --- | --- |
| Class | Attribute of class |
| Quote | RatedQuote, Comment, Message |

**Section 4: Exceptions identified**:

|  |  |
| --- | --- |
| Exception | How it will be handled |
| Submission Failure | Display error message |
| Server error | Display a friendly error page |

### UC09: Report Quote

**Identifier**: UC09

**Brief Description**: This use case enables the user to report quote that they think it is bad or an expletive. It might take up to 24 hours to the administrator to check the quote.

**Section 1**: Business Rule(s):

* BR13: User can report one quote at one time
* BR16: Administrator can check multiple quotes at one time
* BR09: Quote report reason cannot contain special characters or expletives

**Section 2**: Main Scenario with Alternative Flows (HD):

**Main Flow**: Report Quote

**Preconditions**:

The user is on the main page and already signed in.

| Step # | Actor: User | System | Data Used |
| --- | --- | --- | --- |
| 1 | Clicks on the Down Arrow Icon beside the quote |  |  |
| 2 | Clicks on Report Quote option | Displays a pop-up window and prompts to choose the reasons | *Reason* |
| 3 | Clicks on Submit button | Displays a confirm message | *Message* |

**Alternative Flow**

|  | Alternative Flow | Description |
| --- | --- | --- |
| 1 | Report Reason contains special characters or expletives | Display a warning message about the special characters or expletives and return to the main page |

**Section 3: Classes/Attributes identified**:

|  |  |
| --- | --- |
| Class | Attribute of class |
| Report | Reason, Message |

**Section 4: Exceptions identified**:

|  |  |
| --- | --- |
| Exception | How it will be handled |
| Invalid reason | Display message on why the reason is invalid |
| Server error | Display a friendly error page |

### UC10: Approve/Reject Quote

**Identifier**: UC10

**Brief Description**: This use case enables an admin to approve a quote which was uploaded by a user but was marked for admin approval by the system’s quote filter.

**Section 1**: Business Rule(s): N/A

**Section 2**: Scenario with alternative flow, “rejecting a quote”:

**Main Flow**: Approve Quote

**Preconditions**:

The user is on the approve quote page and already signed in with admin privileges.

| Step # | Actor: Admin | System | Data Used |
| --- | --- | --- | --- |
| 1 | Admin clicks on “Pending for approval” | System displays a list of all the quotes which are pending for admin approval. | Quotes with status “pending approval” |
| 2 | Admin clicks on the “Approve” button besides the listing | System changes the status of the quote to **approved** and pushes it to the active quote directory in the database. |  |
| 3 |  | An appropriate message is displayed on the screen and then the page is refreshed to show all the latest quotes needed to be approved. |  |

**Alternative Flow**

| Step # | Actor: Admin | System | Data Used |
| --- | --- | --- | --- |
| 1 | Admin clicks on “Pending for approval” | System displays a list of all the quotes which are pending for admin approval. | Quotes with status “pending approval” |
| 2 | Admin clicks on the “Reject” button besides the listing | System changes the status of the quote to **rejected**. The quote is pushed to the rejected directory. |  |
| 3 |  | An appropriate message is displayed on the screen and then the page is refreshed to show all the latest quotes needed to be approved. |  |

**Section 3: Classes/Attributes identified**:

|  |  |
| --- | --- |
| Class | Attribute of class |
| Quote | status |

**Section 4: Exceptions identified**:

|  |  |
| --- | --- |
| Exception | How it will be handled |
| N/A | N/A |

### UC11: Machine Filter Management

**Identifier**: UC10

**Brief Description**: This use case enables an admin to approve a quote which was uploaded by a user but was marked for admin approval by the system’s quote filter.

**Section 1**: Business Rule(s): N/A

**Section 2**: Main flows:

**Preconditions**:

The admin is on the “Filter management” page and already signed in with admin privileges.

Scenario 1: Adding a new filter string

| Step # | Actor: Admin | System | Data Used |
| --- | --- | --- | --- |
| 1 | Admin clicks on “Add new filter string” | System displays a list of existing filter strings and a textbox to input a new string |  |
| 2 | Admin enters the new filter string and hits “Submit” | System checks if the string is duplicated or very similar to some other string |  |
| 3 |  | An appropriate success message is displayed on the screen. Then the main page for filter management is displayed |  |

Scenario 2: Deleting an existing filter string

| Step # | Actor: Admin | System | Data Used |
| --- | --- | --- | --- |
| 1 | Admin clicks on “Delete a filter string” | System displays a list of existing filter strings with checkboxes besides each entry |  |
| 2 | Admin checks the checkboxes corresponding to the string that are to be deleted and hits the “Delete” button | System displays a list of all the selected filter strings and asks for deletion confirmation |  |
| 3 | Admin clicks on “Confirm” button | An appropriate success message is displayed on the screen. System removes the deleted strings from the database. Then the main page for filter management is displayed |  |

**Section 3: Classes/Attributes identified**:

|  |  |
| --- | --- |
| Class | Attribute of class |
| Filter | filterString |

**Section 4: Exceptions identified**:

|  |  |
| --- | --- |
| Exception | How it will be handled |
| N/A | N/A |

### UC12: Admin Profile Edit

**Identifier**: UC12

**Brief Description:**

This use case enables the admin to change and edit the profile.

**Section 1: Business Rule(s):**

* BR01 - The Username can contain alphanumeric characters and no more than 25 characters.
* BR02 The password can contain alphanumeric characters and no more than 25 characters.
* BR03 Date of birth will be formatted YYYY-MM-DD.
* BR04 Email must be a valid email address format and verified.
* BR08 New username cannot be duplicate
* BR07 Password can be changed only after email verification.

**Section 2: Main Scenario**

**Preconditions:**

The user is on the main page and must be logged in.

| **Step #** | **Actor: User** | **System** | **Data Used** |
| --- | --- | --- | --- |
| 1 | Clicks on Profile button | Redirect to the User Profile |  |
| 2 | View and edit the profile on the left panel | Validate the information according to the business rules | *Username*  *Firstname*  *Lastname*  *Email address*  *Password*  *Date of Birth*  *Position* |
| 3 | Click Save button to save the changes | Save the changes in the database and refresh the page with the new profile |  |

**Alternative Flow**

|  | **Alternative Flow** | **Description** |
| --- | --- | --- |
| 1 | Fails in validating the profile which results in error | This is for when the proper rules according to the business rules |

**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| **Class** | **Attribute of class** |
| User | Username, password, firstname, lastname, email, date of birth |

**Section 4: Exceptions identified:**

|  |  |
| --- | --- |
| **Exception** | **How it will be handled** |
| Internal Server Error | Error page is displayed with the contact information. |

### UC14: User Profile Edit

**Identifier**: UC14

**Brief Description:**

This use case enables the user to change and edit the profile.

**Section 1: Business Rule(s):**

* BR01 - The Username can contain alphanumeric characters and no more than 25 characters.
* BR02 The password can contain alphanumeric characters and no more than 25 characters.
* BR03 Date of birth will be formatted YYYY-MM-DD.
* BR04 Email must be a valid email address format and verified.
* BR08 New username cannot be duplicate
* BR07 Password can be changed only after email verification.

**Section 2: Main Scenario**

**Preconditions:**

The user is on the main page and must be logged in.

| **Step #** | **Actor: User** | **System** | **Data Used** |
| --- | --- | --- | --- |
| 1 | Clicks on Profile button | Redirect to the User Profile |  |
| 2 | View and edit the profile on the left panel | Validate the information according to the business rules | *Username*  *Firstname*  *Lastname*  *Email address*  *Password*  *Date of Birth* |
| 3 | Click Save button to save the changes | Save the changes in the database and refresh the page with the new profile |  |

**Alternative Flow**

|  | **Alternative Flow** | **Description** |
| --- | --- | --- |
| 1 | Fails in validating the profile which results in error | This is for when the proper rules according to the business rules |

**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| **Class** | **Attribute of class** |
| User | Username, password, firstname, lastname, email, date of birth |

**Section 4: Exceptions identified:**

|  |  |
| --- | --- |
| **Exception** | **How it will be handled** |
| Internal Server Error | Error page is displayed with the contact information. |

### UC15: Customize Reminder Settings

**Identifier**: UC15

**Brief Description:**

This use case enables the user to customize the quote reminder settings

**Section 1: Business Rule(s):**

* BR06 Minimum 0, Maximum 5 Preferred alert time can be selected.
* BR10 One user can only select one mind state at one time for reminder
* BR11 One user can only have one email to set the quote reminder
* BR12 User can only set one quote reminder per day on selected week days which repeats every week.

**Section 2: Main Scenario**

**Preconditions:**

The user is on the main page and must be logged in.

| **Step #** | **Actor: User** | **System** | **Data Used** |
| --- | --- | --- | --- |
| 1 | Clicks on Profile button | Redirect to the User Profile |  |
| 2 | View and change the customize reminder settings in the right panel | Validate the information according to the business rules | *Mind State*  *Time*  *Number of Days* |
| 3 | Click Set button to save the changes | Save the changes in the database and refresh the page with the new reminder settings |  |

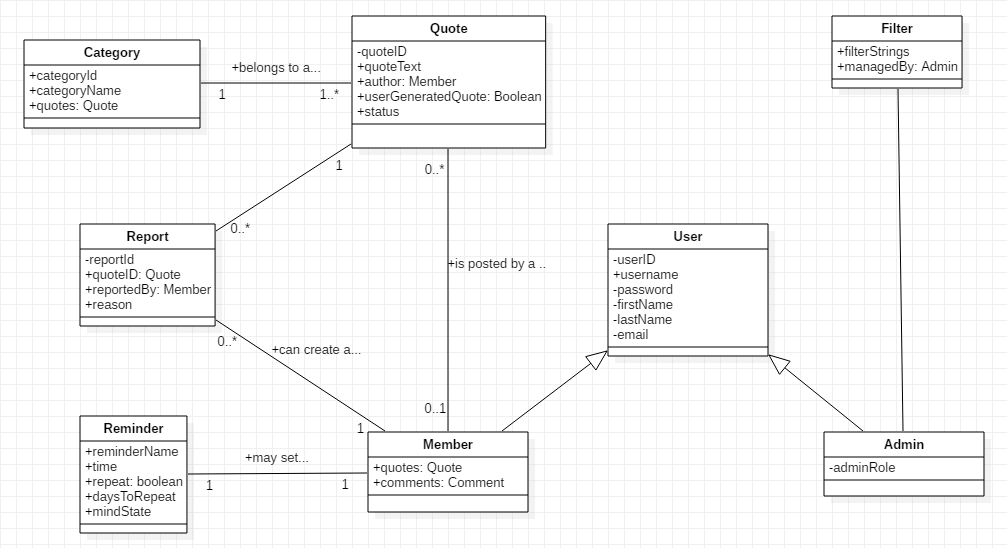
**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| **Class** | **Attribute of class** |
| Reminder | Username, mind state, time, weekdays |

**Section 4: Exceptions identified:**

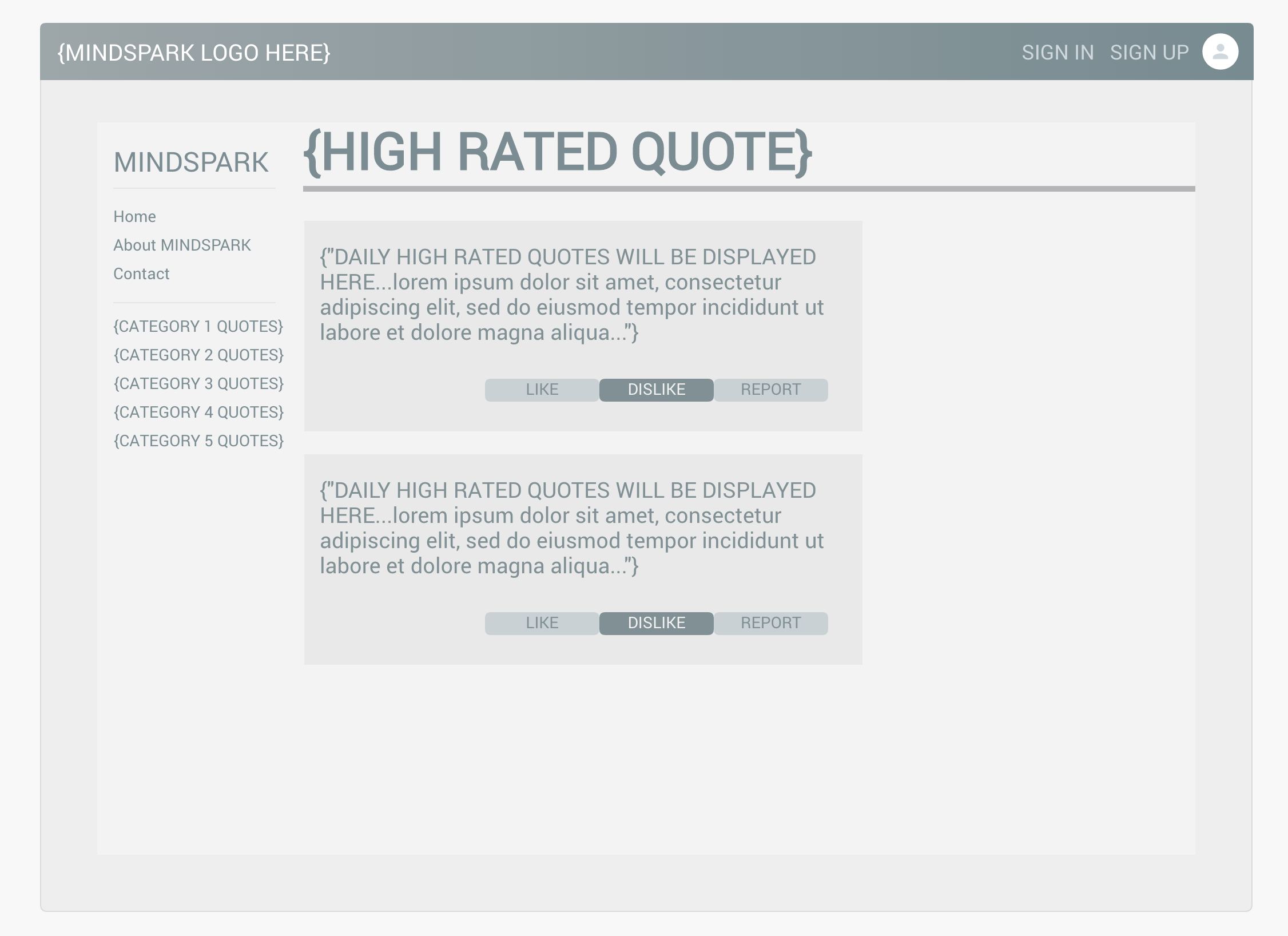
|  |  |
| --- | --- |
| **Exception** | **How it will be handled** |
| Internal Server Error | Error page is displayed with the contact information. |

## 5.04 Domain Class Diagram

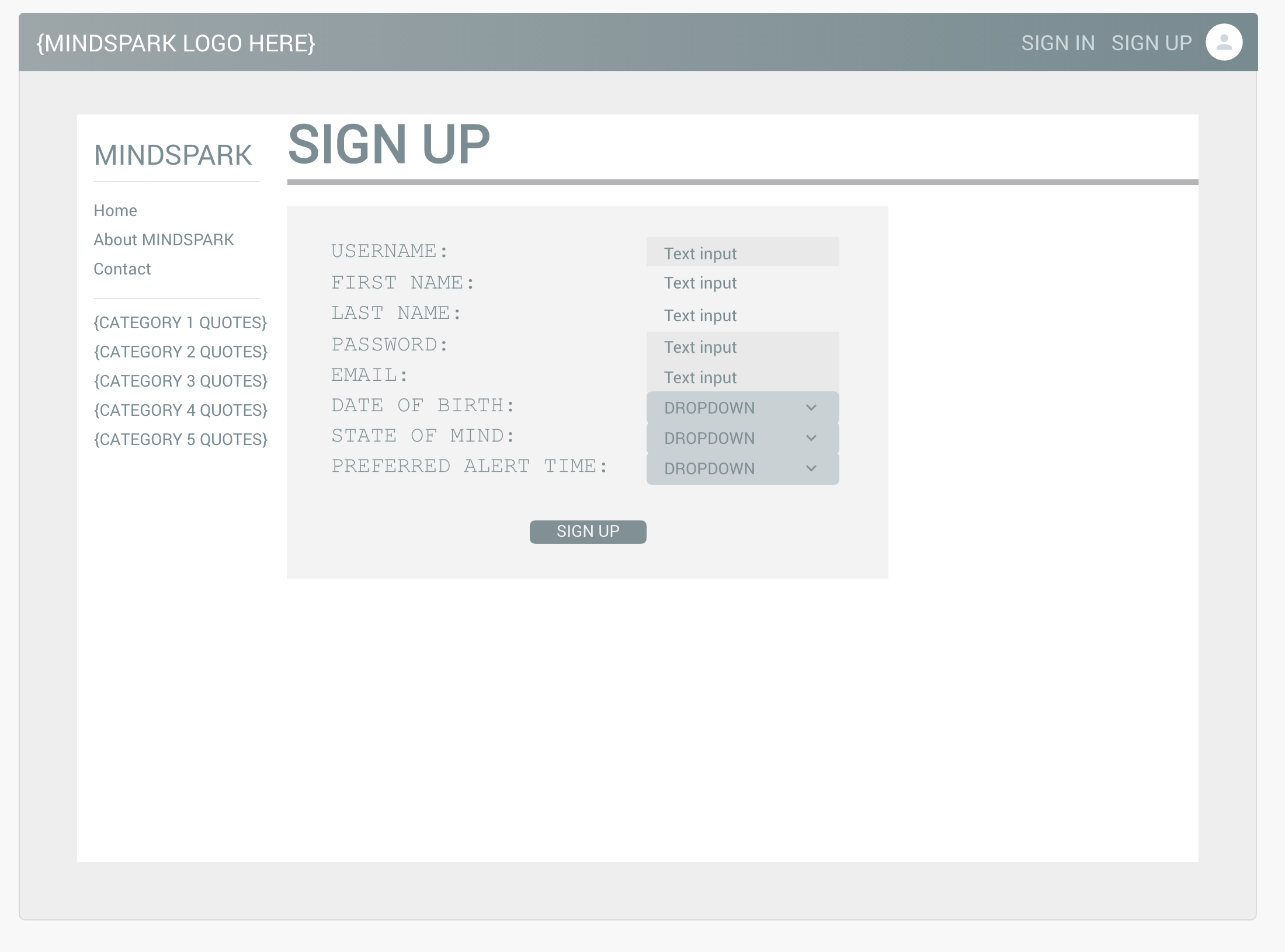


# 6. – Interface Mock-ups

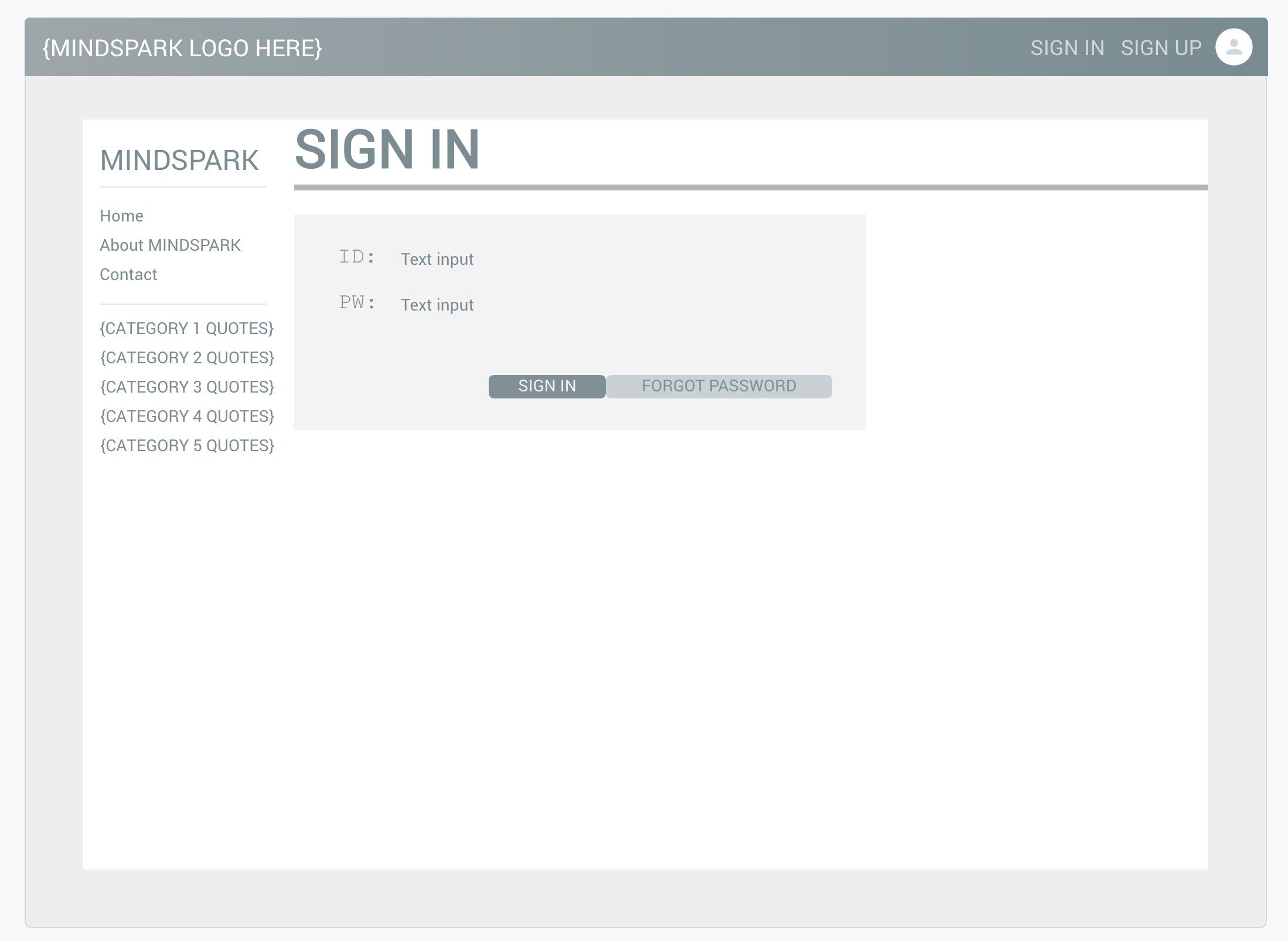
### 6.01 - Main page



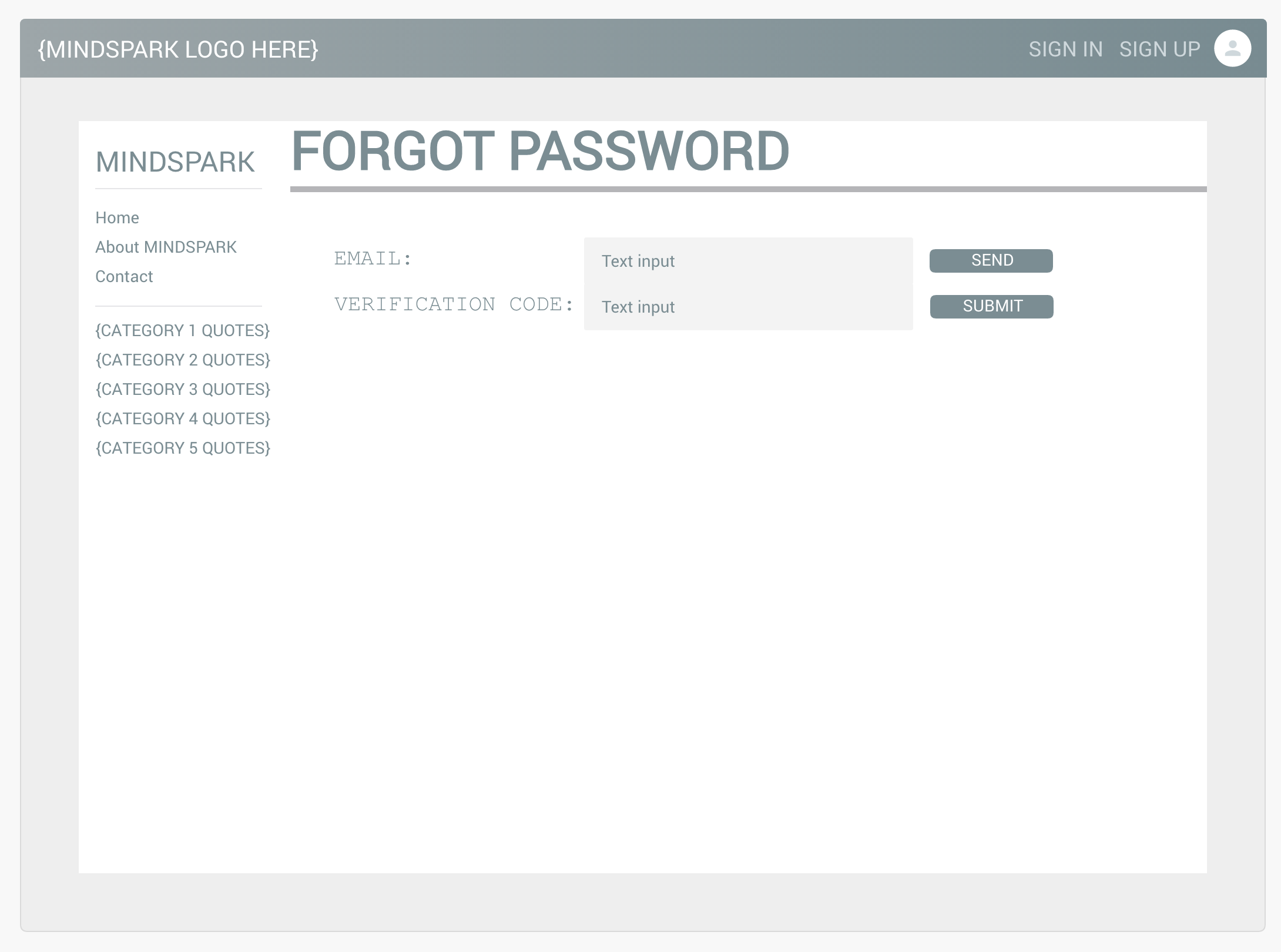
### 6.02 – Sign up



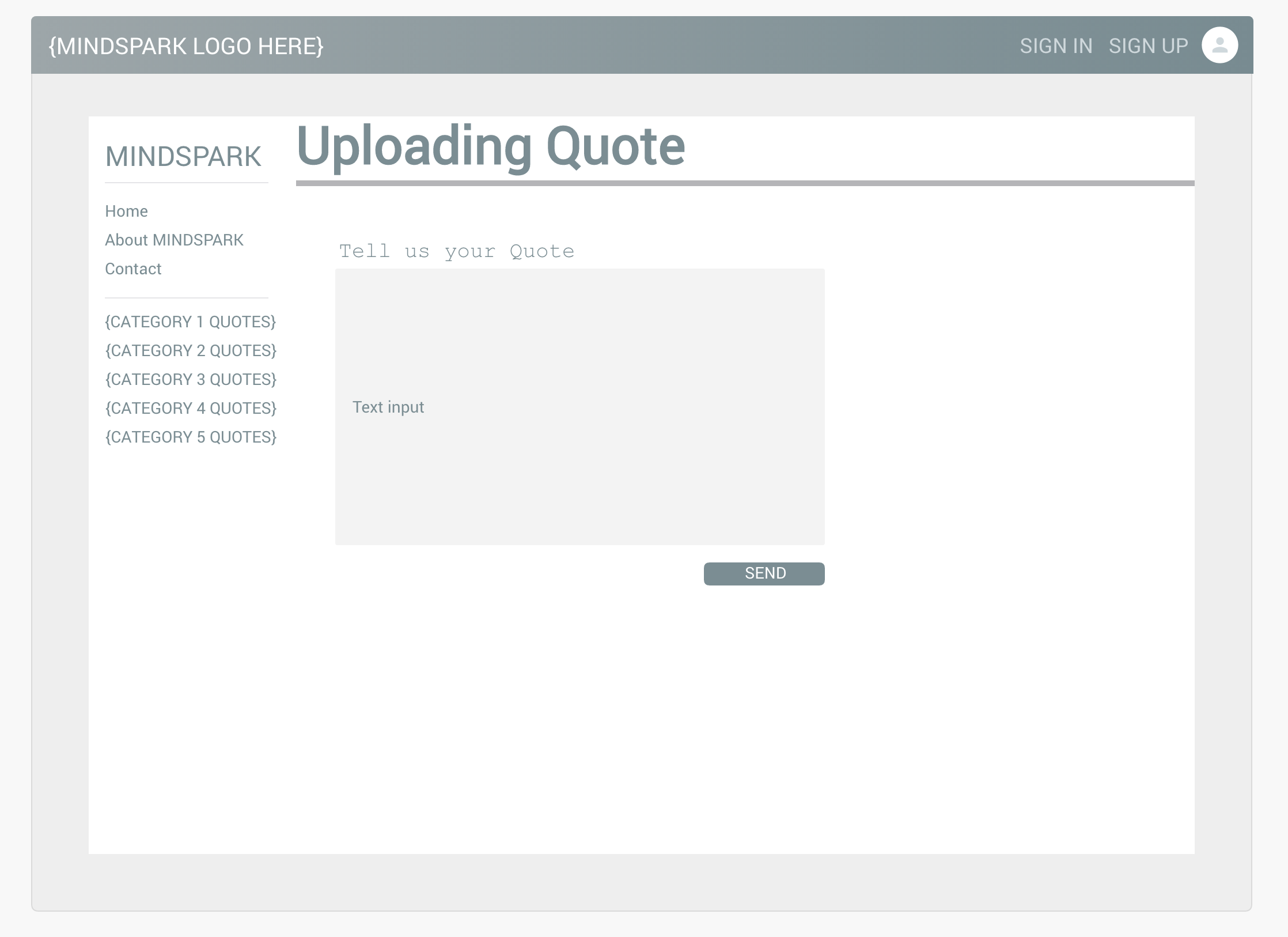
### 6.03 – Sign in



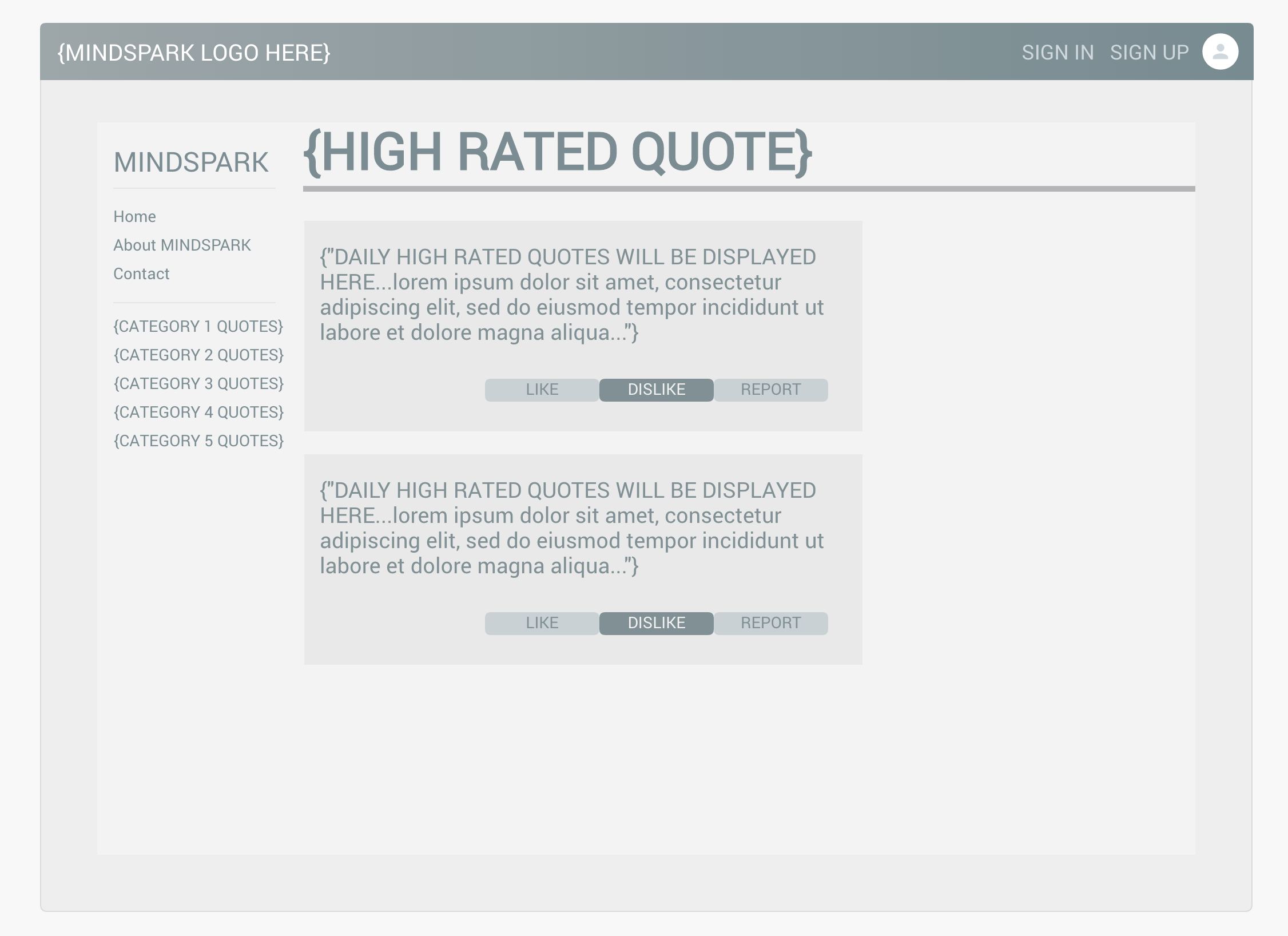
### 6.04 – Forgot Password



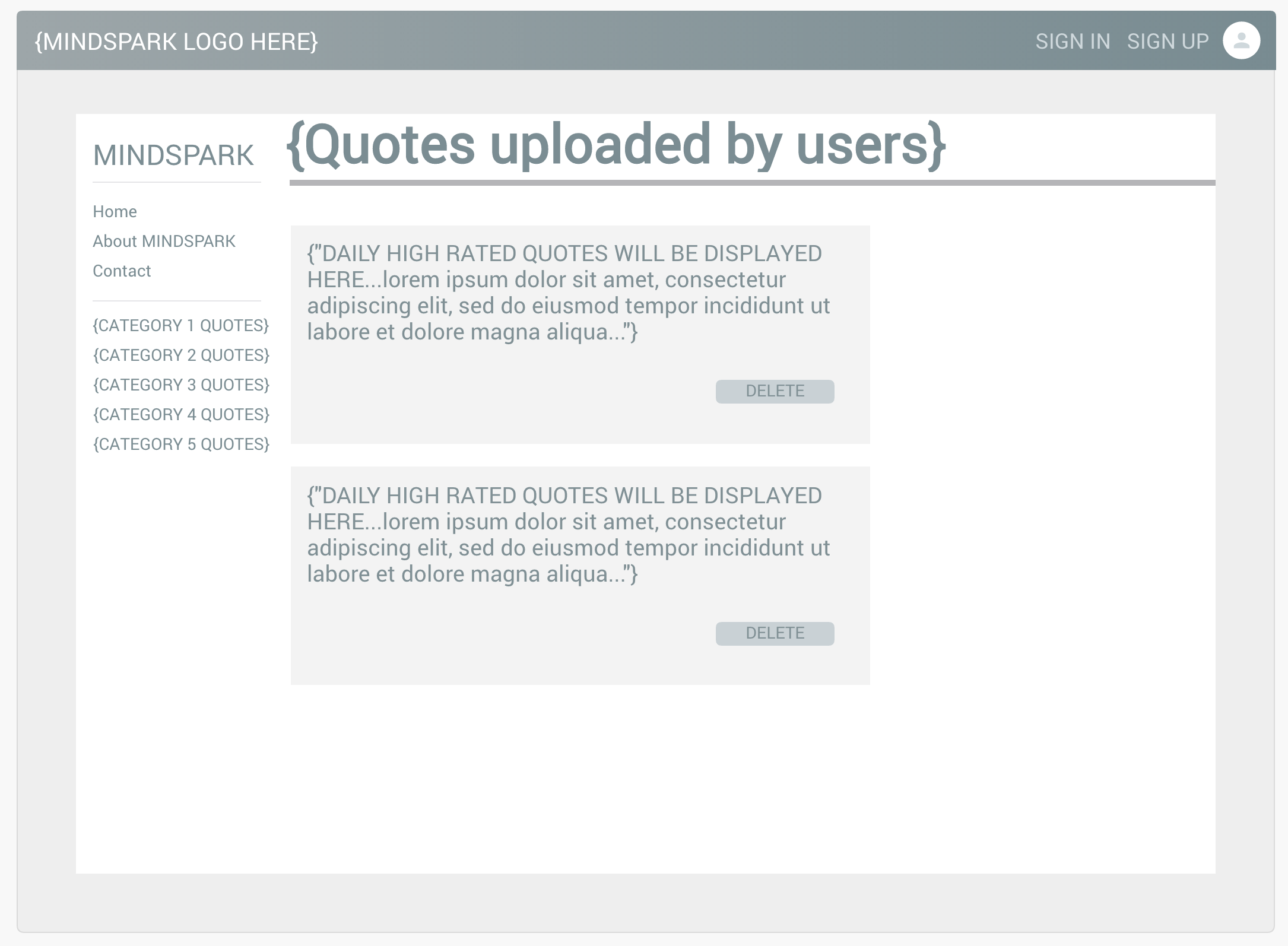
### 6.05 – Upload quote



### 6.06 – Reading quote



### 6.07 – Delete quote



### 6.08 – Rate quote

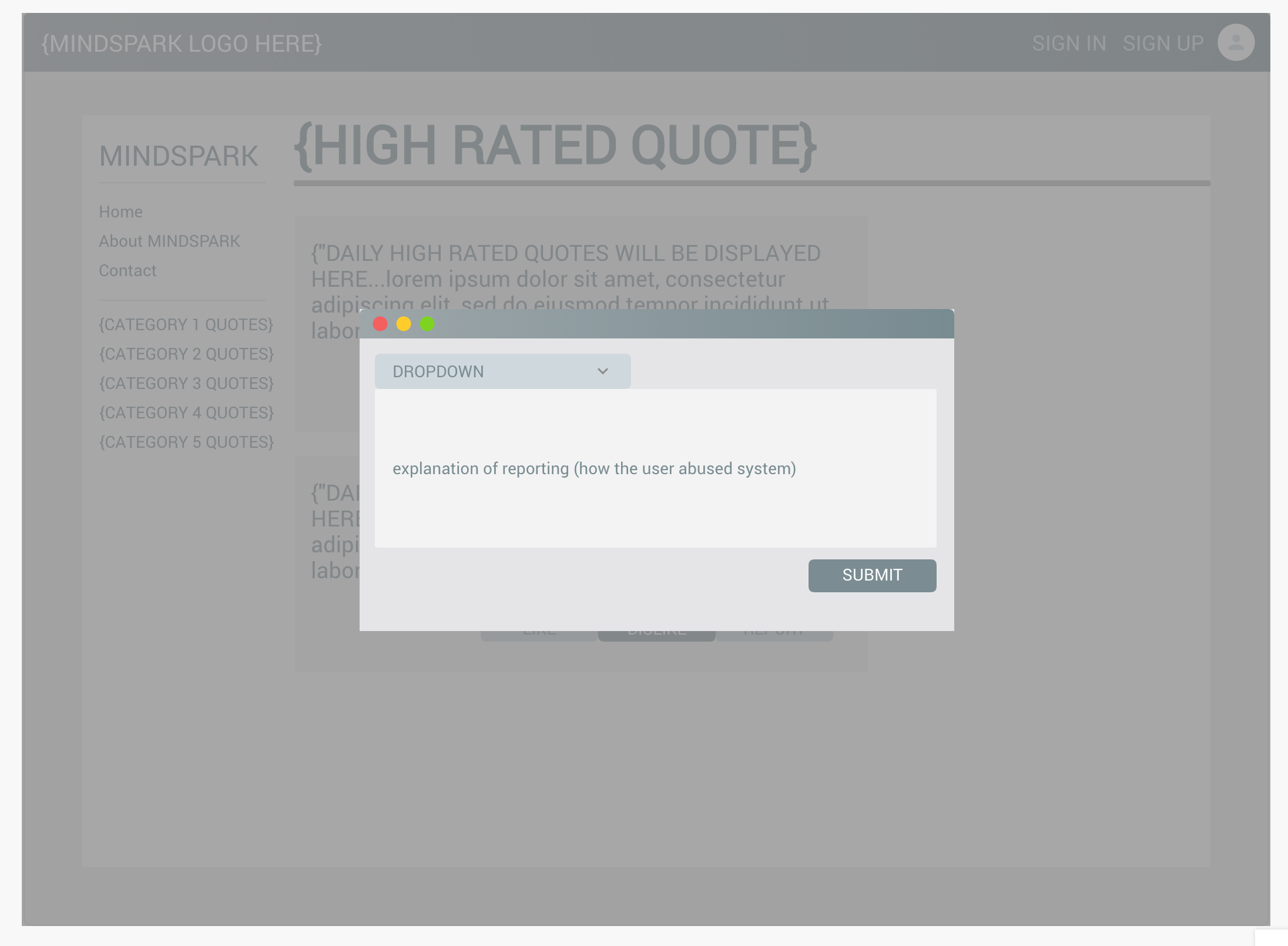
A screenshot of a cell phone

Description automatically generated

### 6.09 – Report quote

A screenshot of a cell phone

Description automatically generated



### 6.10 – Admin profile manage/approve

A screenshot of a cell phone

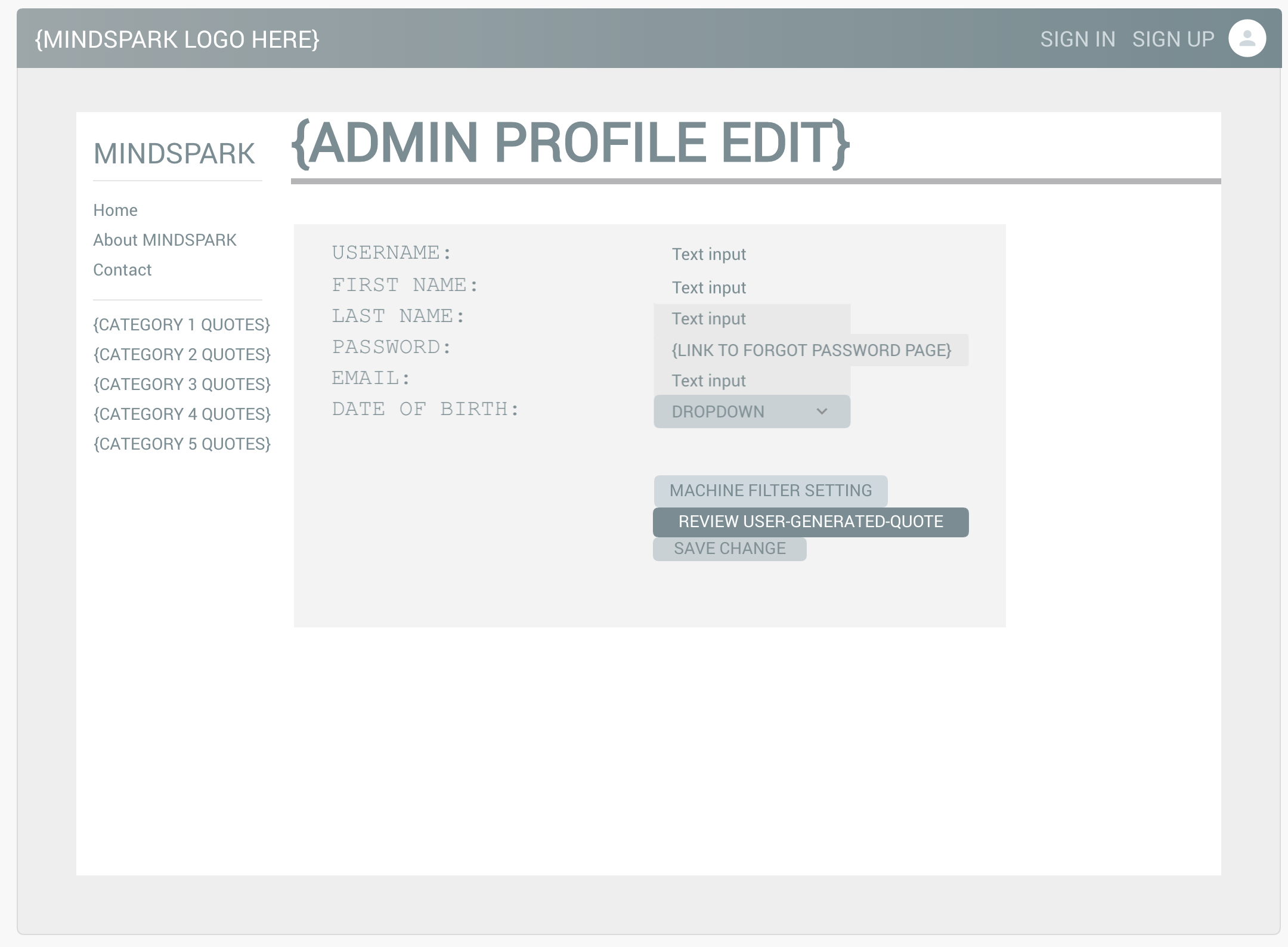
Description automatically generated

### 6.11 – Admin profile manage/reject

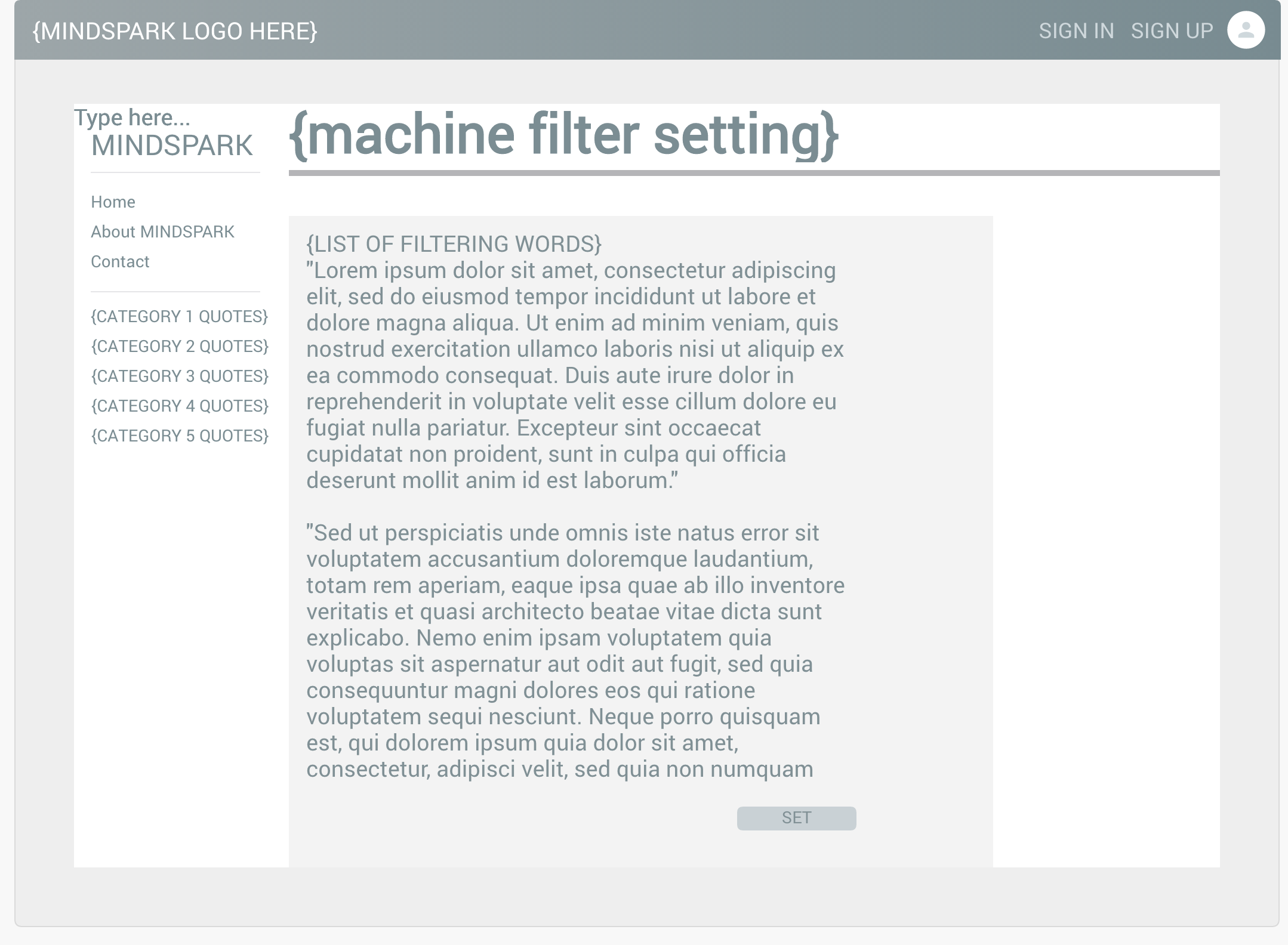
A screenshot of a cell phone

Description automatically generated

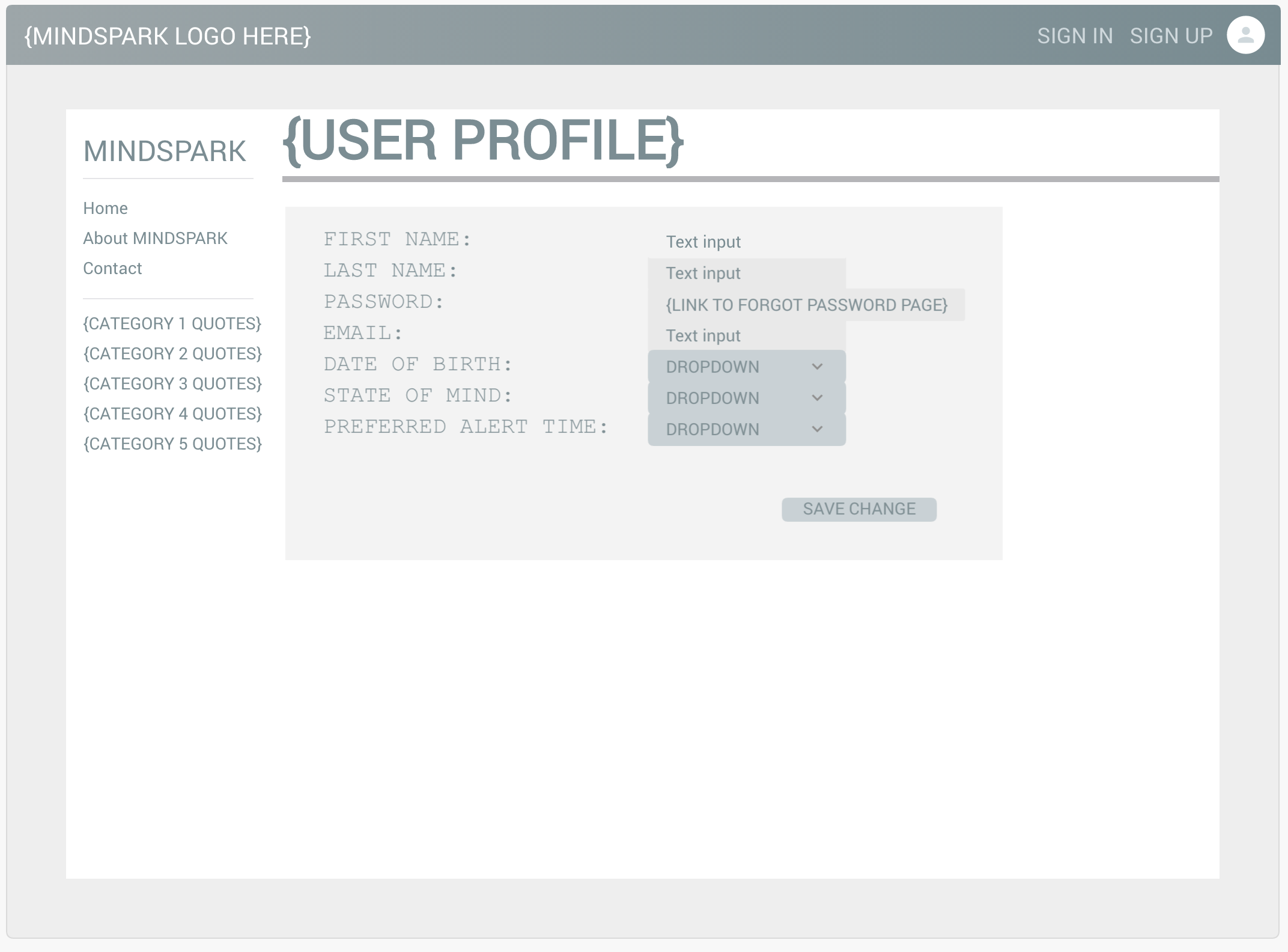
### 6.12 – Admin profile manage/edit



### 6.13 – Admin profile manage/machine filter management



### 6.14 – User profile manage / profile edit



### 6.15 - User profile manage / customize reminder setting

A screenshot of a cell phone

Description automatically generated

# 7.0 – Database

### 7.01 – Scripts to Create Database Tables

#### User Table

db.createCollection("user", {

validator: {

$jsonSchema: {

bsonType: "object",

required: [ "userID", "username", "password", "firstName", "lastName", "email", "quoteIDs", "adminRole" ],

properties: {

userID: {

bsonType: "string",

description: "must be a string and is required"

},

username: {

bsonType: "string",

description: "must be a string and is required"

},

password: {

bsonType: "string",

description: "must be a string and is required"

},

firstName: {

bsonType: "string",

description: "must be a string and is required"

},

lastName: {

bsonType: "string",

description: "must be a string and is required"

},

email: {

bsonType: "string",

description: "must be a string and is required"

},

adminRole: {

bsonType: "string",

enum: ["admin", "moderator", "none"],

description: "must be a string and is required"

},

}

}

}

})

#### Report Table

db.createCollection("report", {

validator: {

$jsonSchema: {

bsonType: "object",

required: [ "reportID", "quoteID", "memberID", "reason" ],

properties: {

reportID: {

bsonType: "string",

description: "must be a string and is required"

},

quoteID: {

bsonType: "string",

description: "must be a string and is required"

},

memberID: {

bsonType: "string",

description: "must be a string and is required"

},

reason: {

bsonType: "string",

description: "must be a string and is required"

}

}

}

}

})

#### Reminder Table

db.createCollection("reminder", {

validator: {

$jsonSchema: {

bsonType: "object",

required: [ "reminderName", "time", "repeat", "days", "mindset" ],

properties: {

reminderName: {

bsonType: "string",

description: "must be a string and is required"

},

time: {

bsonType: "string",

description: "must be a string and is required"

},

repeat: {

bsonType: "boolean",

description: "must be a boolean and is required"

},

days: {

bsonType: [ "string" ],

description: "must be an array of strings and is required"

},

mindset: {

bsonType: [ "string" ],

enum: [ "happy", "sad", "depressed", "stressed" ],

description: "must be a string and is required"

}

}

}

}

})

#### Quote Table

db.createCollection("quote", {

validator: {

$jsonSchema: {

bsonType: "object",

required: [ "quoteID", "quoteText", "memberID", "userGeneratedQuote", "status" ],

properties: {

quoteID: {

bsonType: "string",

description: "must be a string and is required"

},

quoteText: {

bsonType: "string",

description: "must be a string and is required"

},

memberID: {

bsonType: "string",

description: "must be a string and is required"

},

userGeneratedQuote: {

bsonType: "boolean",

description: "can only be true or false and is required"

},

status: {

bsonType: [ "string" ],

enum: [ "approved", "rejected", "waiting", "posted" ],

description: "must be a string and is required"

}

}

}

}

})

#### Category Table.

db.createCollection("category", {

validator: {

$jsonSchema: {

bsonType: "object",

required: [ "categoryId", "categoryName" ],

properties: {

categoryId: {

bsonType: "int",

description: "must be an integer and is required"

},

categoryName: {

bsonType: "string",

description: "must be a string and is required"

}

}

}

}

})

### 7.02 – Scripts to Insert Data(meaningful data)

#### User Table

db.user.insert([

{

username: ‘papanchani1’,

password: ‘t8yuRTe4’,

firstName: ‘Pratik’,

lastName: ‘Panchani’,

adminRole: ‘true’,

email: ‘papanchani1@myseneca.ca’

},

{

username: ‘mason1’,

password: ‘yu4b5et3’,

firstName: ‘Mason’,

lastName: ‘Ko’,

adminRole: ‘true’,

email: ‘masonk@myseneca.ca’

}

])

#### Report Table

db.report.insert([

{

reportID: 1,

quoteID: 1,

reportedBy: ‘papanchani1’,

reason: ‘expletive word’

},

{

reportID: 2,

quoteID: 1,

reportedBy: ‘masonk’,

reason: ‘racial discrimination’

}

])

#### Reminder Table

db.reminder.insert([

{

username: ‘papanchani1’,

reminderStatus: ‘ON’,

time: 18,

days: [ “Monday”, “Tuesday”, “Wednesday”, “Thursday”, “Friday”]

categoryID: 1

},

{

username: ‘masonk’,

reminderStatus : ‘ON’,

time: 9,

days: [ “Monday”, “Tuesday”, “Thursday”]

categoryId: 3

}

])

#### Quote Table

db.quote.insert([

{

quoteID: 1,

quoteText: ‘Quote text’,

author: ‘papanchani1’,

userGeneratedQuote: ‘true’,

status: ‘waiting,

categoryID: 1

},

{

quoteID: 2,

quoteText: ‘Quote text’,

author: ‘masonk’,

userGeneratedQuote: ‘true’,

status: ‘PUBLIC’,

categoryID: 4

}

])

#### Category Table

db.category.insert([

{

categoryID: 1,

categoryName: ‘Stressed’

},

{

categoryID: 2,

categoryName: ‘Depressed’

},

{

categoryID: 3,

categoryName: ‘Sad’

},

{

categoryID: 4,

categoryName: ‘Happy’

}

])

### 7.03 – Scripts to Delete Tables:

db.user.drop()

db.quote.drop()

db.category.drop()

db.report.drop()

db.reminder.drop()

### 7.04 – Scripts to populate tables (Alternate SQL queries)

**User table**

INSERT INTO `User` (`USERNAME`, `PASSWORD`, `FIRST\_NAME`,

`LAST\_NAME`, `ADMIN\_ROLE`, `,

`DATE\_OF\_BIRTH`,`EMAIL`) VALUES

('papanchani1', 't8yuRTe4', 'Pratik', 'Panchani','Y',

'1998-01-31','papanchani1@myseneca.ca),

('mason1', 'yu4b5et3', 'Mason', 'K', 'Y',

'1998-07-03', 'en', 'masonk@myseneca.ca');

**Quote table**

INSERT INTO `Quote` (`QUOTE\_ID`, `QUOTE\_TEXT`, `QUOTE\_AUTHOR`, `USER\_GENERATED`,`STATUS`,`CATEGORY\_ID`) VALUES

(1, 'Quote text 1', 'papanchani1', 'Y','PUBLIC','1'),

(2, 'Quote text 2', 'masonk', 'Y','PUBLIC','4');

**Category table**

INSERT INTO INTO `Category` (`CATEGORY\_ID`, `CATEGORY\_DESCRIPTION`) VALUES

(1, 'Stressed'),(2, 'Depressed'),(3, 'Sad');

**Report table**

INSERT INTO INTO `Report` (`REPORT\_ID`, `QUOTE\_ID`,`REPORTED\_BY`,`REASON`) VALUES

(1,1,'papanchani1','expletive word'),(2,1,masonk,'racial discrimination');

**Reminder table**

INSERT INTO INTO `Reminder` (`USERNAME`, `REMINDER\_STATUS`,`TIME`,`MONDAY`,`TUESDAY`,

`THURSDAY`,`FRIDAY`,`SATURDAY`,`SUNDAY`,`CATEGORY\_ID`) VALUES

('papanchani1','ON',18:00,'Y','Y','Y','Y','Y','N','N',1),

('masonk','ON',09:00,'N','N','Y','Y','N','Y','Y',3);

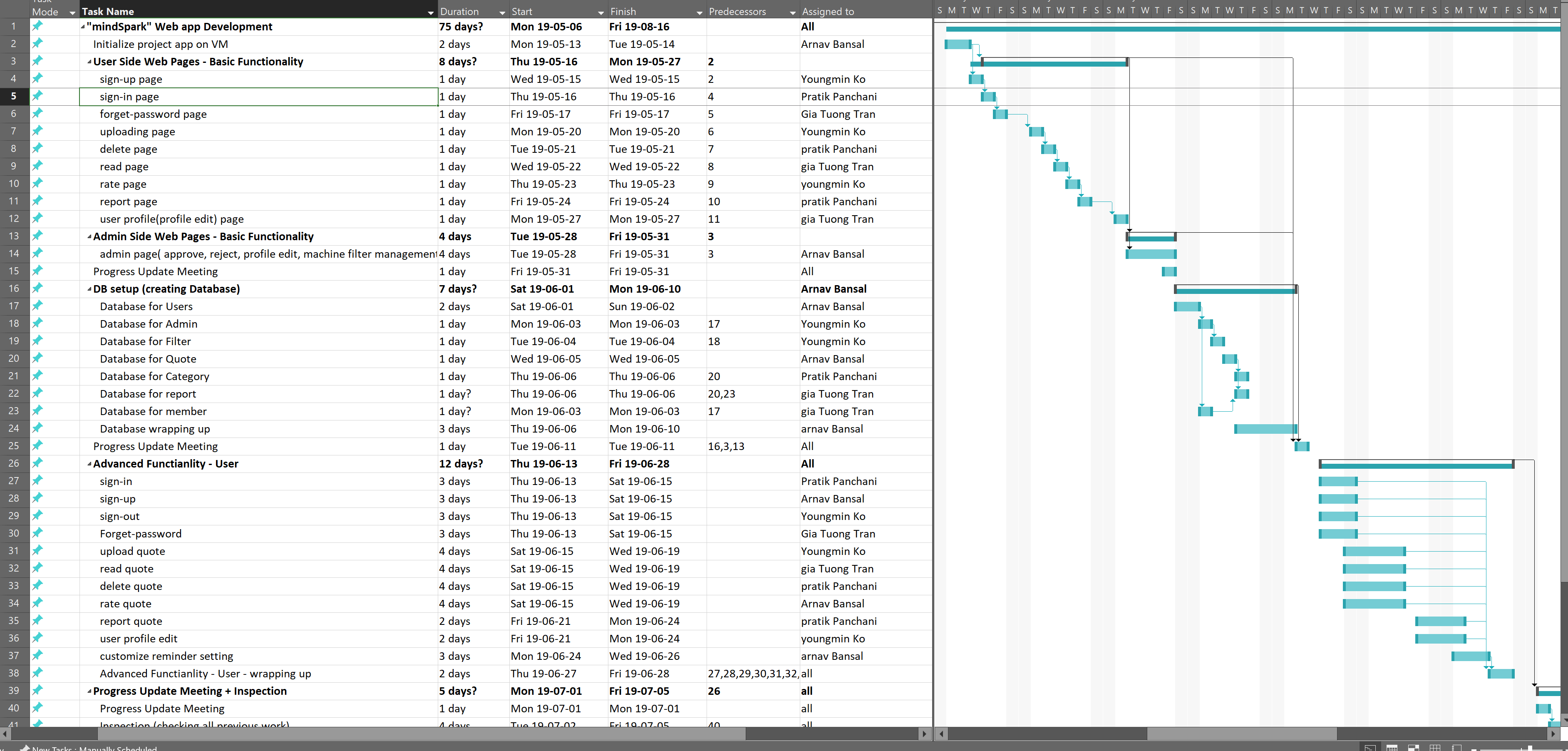
### 7.05 – Data Dictionary

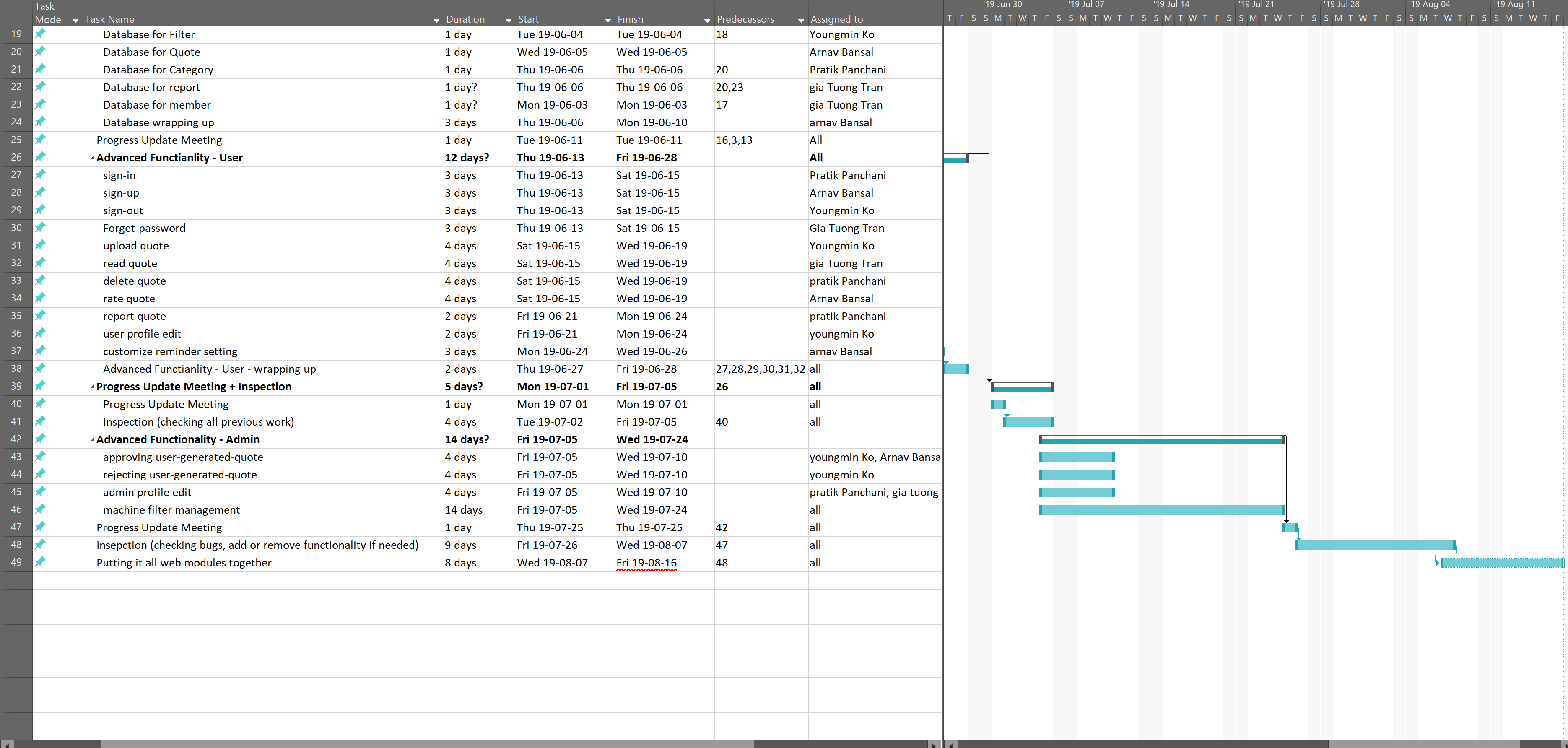
Note: The field lengths mentioned here will not have a corresponding constraint that will be applied on the database or tables. The lengths of the fields will be enforced programmatically within the application front-end.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Table** | **Data Type** | **Length** | **Read Only?** | **NULL** | **Required** | **Description** |
| userID | user | VARCHAR | 25 | Yes | No | Yes | Auto-generated unique id. Assigned to a user account on signup |
| username | user | VARCHAR | 25 | No | No | Yes | Username for a user account. Needed for login. |
| password | user | VARCHAR | 40 | No | No | Yes | A password for a user account. Needed for login. Only an encrypted version of the password will be stored on the database. |
| firstname | user | VARCHAR | 40 | No | No | Yes | The first name for the person holding a user account |
| lastname | user | VARCHAR | 40 | No | No | Yes | The last name for the person holding a user account |
| email | user | VARCHAR | 100 | No | No | Yes | The email associated with a user account |
| quoteIDs | user | Array of strings | N/A | No | No | Yes | An array of strings containing the quoteID of all the quotes posted by a particular user |
| adminRole | user | VARCHAR | 20 | Yes | No | Yes | A field indicating the authorization level of a user account |
| reportID | report | VARCHAR | 20 | Yes | No | Yes | A unique ID for a report created when reporting a quote |
| quoteID | report | VARCHAR | 20 | Yes | No | Yes | The unique id for the quote being reported |
| memberID | report | VARCHAR | 20 | Yes | No | Yes | The unique id of the member creating the report |
| reason | report | VARCHAR | 120 | No | No | Yes | A string message indicating the reason a quote is being reported |
| reminderName | reminder | VARCHAR | 20 | No | No | Yes | An identifying name for a reminder set by the user |
| time | reminder | TIMESTAMP | N/A | No | No | Yes | The time for which the reminder is set |
| repeat | reminder | Boolean | N/A | No | No | Yes | A boolean representing whether the reminder is to be repeated |
| days | reminder | Array of strings | N/A | No | No | Yes | A list of days when the reminder will be repeated |
| mindset | reminder | VARCHAR | 20 | No | No | Yes | The mindset of the user for a reminder. Set by the user. |
| quoteID | quote | VARCHAR | 20 | Yes | No | Yes | A unique identifying id for a quote. Auto-generated when a new quote is posted |
| quoteText | quote | VARCHAR | 20 | No | No | Yes | The text body of the quote |
| memberID | quote | VARCHAR | 20 | No | No | Yes | A unique identifying id for the member who posted the quote |
| userGeneratedQuote | quote | Boolean | N/A | No | No | Yes | A boolean value to determine if the posted quote was posted by a member or auto-posted by the system |
| status | quote | VARCHAR | N/A | No | No | Yes | A string value indicating the processing status of a new quote |
| categoryID | category | VARCHAR | 20 | Yes | No | Yes | A unique identifying id for category. Auto-generated when a new a category is added |
| categoryName | category | VARCHAR | 20 | No | No | Yes | The name of the category |

# 8. – Implementation Schedule for PRJ666 (week 12)

## 8.01 – Schedule (Make sure to include names, dependencies, start and finish dates and hours)





**More information in actual Microsoft Project Document**

# 9 – Measurable deliverables

1. User Functionality
   1. Login, navigate and interact with the application
   2. Post a new quote
   3. Rate a quote
   4. Report a quote
   5. Request for a password change
   6. View the author of a quote
   7. Delete quotes that were posted by the user themselves
   8. View their activity log
2. Admin Functionality
   1. View and modify own profile elements:
      1. Email
      2. Password
      3. Admin role
   2. Be able to reset password through forgot password
   3. View all activity logs (system and user)
   4. View the author of a quote
   5. Manage quote filter
      1. Add, delete, update filter strings
      2. Change number of string matches for auto rejecting a quote
      3. Add exceptions for certain users
   6. Approve and reject new quotes
   7. Disable, enable or ban a user profile
3. Platform
   1. Available as a web application

# 10 – Acceptance Criteria

1. Test application’s responsive web design
2. Test signing up enrollment process
3. Test admin and user account moderation process
4. Test upload functionality
5. Test forgot password functionality
6. Test machine filter functionality for bad input
7. Test admin and user role access privileges
8. Test admin approval process
9. Test logging process
10. Test report functionality
11. Test reminder email notification functionality

# 11 – Client / Faculty Sign-off



# 12 - PRJ666 Virtual Machine (VM) Requirements

**Make sure to include all services – port numbers required to run any service must be assigned by ITS**

Group: “MindSpark” Group Number: 05

Operating System: 1. Linux Ubuntu Server

2. Windows Server

3. Windows 10

4. Other (specify)  **Check & Complete on page two**

Database: 1. MySQL

2. MSSQL

3. MongoDB

4. AZURE

5. Other (specify)  **Check & Complete on page two**

Services: 1. http

2. https

3. ssh

4. RDS (Remote Desk)

5. PhpMyAdmin

6. NodeJS

7. Other (specify)  **Check &** **Complete on page two**

Programming Languages / Framework / Technologies

1. AngularJS \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. JavaScript \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. HTML \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. CSS \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Other information**

**Operating System:**

**Data Base:**

**Services:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**