

# Community Board

## Software Design Document

### Team 4:

Brittany

Brandon

George

Edgar

---

<b>Software Design Document (SDD)</b>	<b>3</b>
1. Introduction	3
1.1 Goals and Objectives	3
Functional requirements	3
Non-Functional requirements	3
Inverse requirements	3
1.2 Statement of Scope	3
1.3 Software Context	4
1.4 Major Constraints	4
<b>2.0 Data Design</b>	<b>4</b>
2.1 Internal Software Data Structures	4
2.2 API Driven Architecture (Globally)	4
2.3 Temporary Data Structure	4
Repository Design Pattern	5
Dependency Injection Design Pattern	5
2.4 Database Description	5
<b>3.0 Architectural and Component-Level Design</b>	<b>6</b>
Architecture Diagram	6
MVC Design Pattern	6
4.0 User Interface Design	7
4.1 Description of the User Interface	7
4.1.1 Screen Images	8

Home Page	8
Register & Login Page/Form	8
Contact/Report Page	9
Create/Update Announcement Page	10
Manage Announcements	11
4.1.2 Objects and Actions	11
4.2 Interface Design Rules	12
4.3 Components Available	12
<b>5.0 Restrictions, Limitations, and Constraints</b>	<b>12</b>
<b>6.0 Testing Issues</b>	<b>13</b>
6.1 Classes of Tests	13
6.2 Expected Software Response	13
6.3 Performance Bounds	13
6.4 Identification of Critical Components	13
<b>7.0 Appendices</b>	<b>13</b>
7.1 Packaging and Installation Issues	13
7.2 References/Supplementary Information	13

# Software Design Document (SDD)

## 1. Introduction

### 1.1 Goals and Objectives

Our goal is to develop an announcement website for people to offer their services such as sales, tutoring, job offering, school-related events. Users will be able to contact the person of the announcement and report inappropriate behavior.

#### **Functional requirements**

- The user will register an account and log in. If login is invalid, an error message will display.
- Logged in users can create, update, and delete their announcements.
- Users will be able to view and search announcements made.
- If no announcements are found given the user's criteria, a "Not Found" message will be displayed.
- Users will be able to contact the person of the post via filling an email form. If the user is not logged in, the system will redirect them to the register page.
- Anyone will be able to report inappropriate announcements and will notify administrators.

#### **Non-Functional requirements**

- Privacy protection (not sharing private information).
- Quick response time when searching/loading posts.
- Error messages/handling, no crashes of server.
- Users must have an internet connection.

#### **Inverse requirements**

- Website doesn't have its own chat system. Customers must fill the contact form and communicate through email.
- Website does not keep track of announcement logs. Once a post has been deleted or been marked as "closed," it will disappear forever.
- Website will not have admin controls to see reports. When an announcement is reported, it will be saved in the database and notify the administrators via email for further action.
- Administrators will review and remove or clear the reported announcement.

### 1.2 Statement of Scope

Upon opening the website, the user will see the home page in which recent announcements will display. Users will either search for announcements of their interest or register an account in order to be able to post announcements in the board. Once registered, the user will have access to the posting form and be able to publish. On the other hand, any customer

interested in any announcement service may click the “contact” button. If not registered, the system will ask the customer to make an account in order to access the contact form. After filling the contact form, an email will be sent to the author of the announcement and the communication goes on from there. Once the customer and author come to an agreement, the author will be able to mark the announcement as “Closed,” which will be deleted after some time.

### 1.3 Software Context

Due to the COVID-19 pandemic, bulletin boards are not in use anymore. Schools closed, people are at home, and school events/clubs are not as known. Very few events are sent through the school emails which not many people check. For those reasons, our team has decided to make these drastic changes alive again.

The purpose of this website is to provide our community a virtual bulletin board (similar to what OCC had in the walls before the pandemic). This is highly useful for those looking to provide their service such as tutoring, those looking for a job, club, or event to join/apply, or even people who might be graduating and want to sell their books or school supplies for a cheaper price.

### 1.4 Major Constraints

- Internet access required for the whole system.
- Database access.
- REST API calls using HTTP protocol.

## 2.0 Data Design

### 2.1 Internal Software Data Structures

Program will create objects (models) for database entries, DTOs (Data Transfer Objects) for sending data to the user/client through our main API. Arrays & lists may be used for storing data collections from API calls.

### 2.2 API Driven Architecture (Globally)

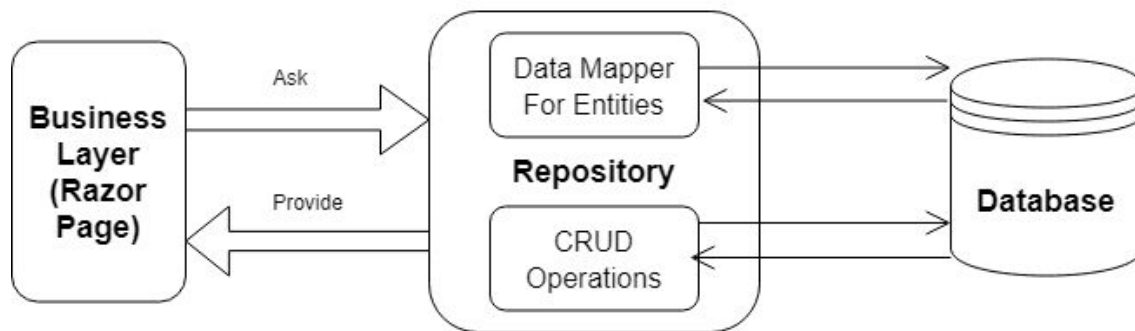
REST API will be used throughout the whole application to perform CRUD (Create, Read, Update, Delete) operations.

### 2.3 Temporary Data Structure

A temporary file may be used to store user announcement reports for administrators to check.

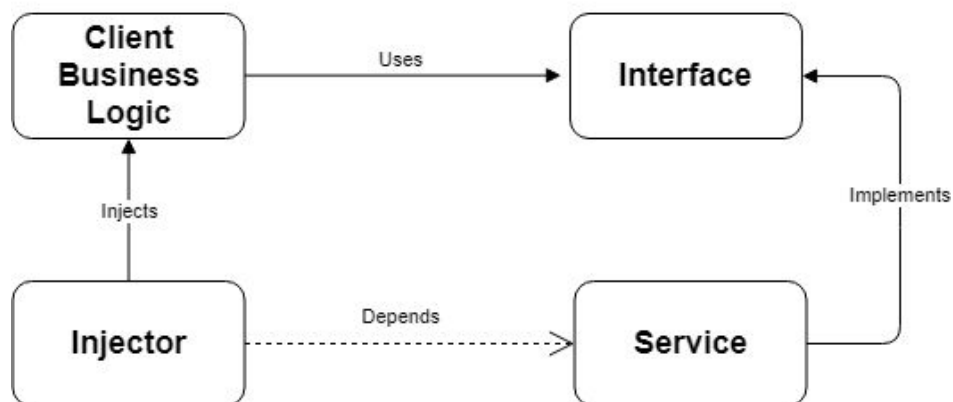
## Repository Design Pattern

### Repository Pattern



## Dependency Injection Design Pattern

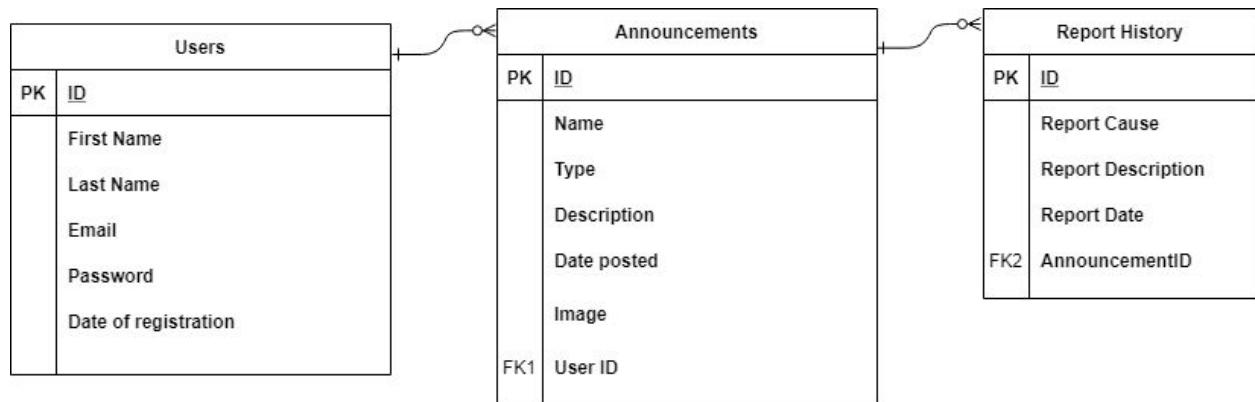
### Dependency Injection Pattern



## 2.4 Database Description

Database contains information about a user (name, username, password), announcements (name, posted date, user, description, etc.), and report history.

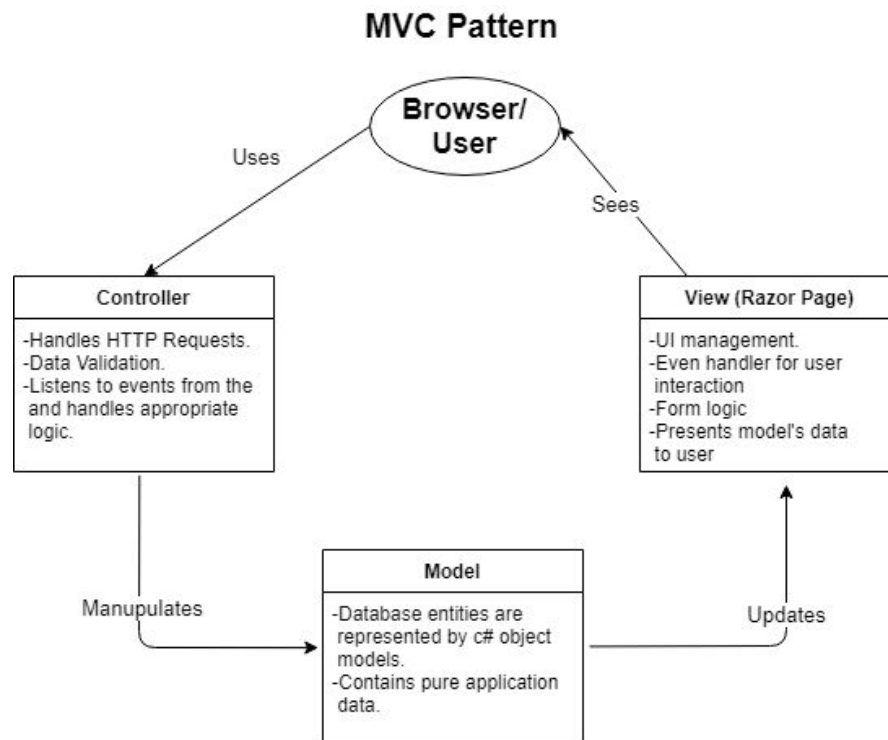
## Entity Relationship Diagram



## 3.0 Architectural and Component-Level Design

### Architecture Diagram

#### MVC Design Pattern



## 4.0 User Interface Design

### 4.1 Description of the User Interface

The user interface is designed using HTML and CSS with Razor Pages in ASP.NET Core 3.1. CSHTML files are used for each razor page allowing to embed C# code into HTML when designing our different application pages.. Responsive design is handled by Bootstrap classes.

Main Pages:

- Home Page (Announcements View)
- Register/Login Page
- Contact Page
- Report Page
- Create Announcement Page
- Manage Announcements Page (For editing/deleting)
- Update Announcement Page

## 4.1.1 Screen Images

### Home Page

# Welcome to the Community Board!



#### Announcement 1

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Ut eum similique repellat a laborum, rerum voluptates ipsam eos quo tempore iusto dolore modi dolorum in pariatur. Incidunt repellendus praesentium quae!

[Contact](#)

#### Announcement 2

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Ut eum similique repellat a laborum, rerum voluptates ipsam eos quo tempore iusto dolore modi dolorum in pariatur. Incidunt repellendus praesentium quae!

[Contact](#)

#### Announcement 3

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Ut eum similique repellat a laborum, rerum voluptates ipsam eos quo tempore iusto dolore modi dolorum in pariatur. Incidunt repellendus praesentium quae!

[Contact](#)

### Register & Login Page/Form

#### Register

Please fill in this form to create an account!



I accept the [Terms of Use](#) & [Privacy Policy](#)

[Register](#)

Already have an account? [Login here](#)

#### Login

Login to start posting or contact!

[Log In](#)

Don't have an account? [Register here](#)



## Contact/Report Page

### Contact (Author's Name Here)

---

Subject

Email Body

Send

### Report Announcement

---

Report Reason

Report Body

Report

## Create/Update Announcement Page

### Create Announcement

-- Announcement Type --

Picture:

No file chosen

Description

Send

### Update Announcement

-- Announcement Type --

Picture:

No file chosen

Description

Send

## Manage Announcements

### Manage Announcements for Specific User

Name	Description	Edit	Delete
Announcement 1	Lorem ipsum dolor sit amet consectetur, adipisicing elit. Ut eum simil ...		
Announcement 2	Lorem ipsum dolor sit amet consectetur, adipisicing elit. Ut eum simil ...		
Announcement 3	Lorem ipsum dolor sit amet consectetur, adipisicing elit. Ut eum simil ...		

### 4.1.2 Objects and Actions

- Home Page:
  - Announcement View: displaying data of the announcement.
  - Contact Button: in each announcement, the user will be able to contact the author.
  - Report Button: in each announcement, the user will be able to report the announcement.
- Register Page:
  - First/Last Name Input: The user will enter their name.
  - Username Input: The user will enter their username.
  - Email Input: The user will enter their email.
  - Password Input: Where the user enters their password.
  - Confirm Password Input: Users must fill this input for password validation.
  - Terms and Conditions Checkbox: User must agree with the terms to be able to register
  - Register Button: User click button to register.
  - Login here link: If the user has already an account, they might want to navigate to the login page
- Login Page:
  - Username/Email Input: The user will enter their username or email to be able to login.
  - Password Input: Where users enter their password for validation and access to their account.
  - Login Button: User click button to login.
  - Register Here Link: If the user does not have an account they might want to create one using this link.

- Contact/Report Page:
  - Subject/Report Reason Input: Input for users to define the subject of email or reason of reporting the announcement.
  - Email/Report Body Textarea: User will type the body of their text here.
  - Send/Report Button: User button to send the request to the server.
- Create/Update Announcement Page
  - Name Input: Announcement name.
  - Announcement-Type Dropdown: User will specify the type of announcement this is.
  - Upload Picture File Input: User input to select image (PNG, JPEG).
  - Description Textarea: Input for describing the announcement.
  - Post Button: Button to send the data to the server.
- Manage Announcements Page:
  - Table: This table shows the name, description, and 2 buttons.
  - Edit Button/Icon: Button to let the user edit their announcement.
  - Delete Button/Icon: Button to let the user delete their announcement.

## 4.2 Interface Design Rules

- Simple and easy to use user interface.
- Responsive design.
- Easy to navigate through pages.

## 4.3 Components Available

- HTML:
  - Forms (For sending requests/responses to server)
  - Inputs (Provides user interaction)
  - Buttons (Clickable control)
  - Labels (For description of components)
  - Textareas (For user input)
  - Other HTML tags (select, sections, headers, footers, navs, etc.)

## 5.0 Restrictions, Limitations, and Constraints

- Mobile view might be a problem. Responsiveness must be achieved when designing pages.
- REST API response time. API should have good performance.
- Web Page load speed.
- User internet's connection.

## 6.0 Testing Issues

### 6.1 Classes of Tests

- Unit Testing: test that each function performs according to specifications.
- Functional Testing: test website components work as expected.
- Performance Testing: test api calls response times and database access.
- User Interface Testing: test UI usability.

### 6.2 Expected Software Response

The software will pass or fail the tests.

### 6.3 Performance Bounds

- Server performance issues are expected.
- Server crashes are definitely expected during development.
- RAM/in-memory database access performance/speed issues expected.

### 6.4 Identification of Critical Components

No critical components are identified as of now.

## 7.0 Appendices

### 7.1 Packaging and Installation Issues

A browser is required to be able to open the webpage (any browser).

### 7.2 References/Supplementary Information

<https://www.geeksforgeeks.org/mvc-design-pattern/>

<https://stackify.com/dependency-injection/>

<https://www.c-sharpcorner.com/UploadFile/b1df45/getting-started-with-repository-pattern-using-C-Sharp/>