CS3354 Software Engineering
Final Project Deliverable 1

ClassMe+

Group members: Alex Armstrong, Daniel Kiv, Hareesh Parchuri, Razvan Preotu, Akhil Gangidi, Nga (Annie) Vu, Craig Hilby

Project Proposal

Web Based Education-Communication Platform (Temporary Name)
Group Members: Alex Armstrong, Razvan Preotu, Nga (Annie) Vu, Akhil Gangidi, Craig Hilby, Hareesh Parchuri, Daniel Kiv

What will you be doing?
We will be doing a web based communication platform that will be used for educational purposes. It included a whiteboard feature so an instructor or student can demonstrate something to the other users on the platform.

Motivation
There is a lack of online educational tools that make teaching simple remotely. Helps teachers give a hands on approach especially for math, science, and language applications where it is helpful to write things out on a whiteboard.

Task Delegation
Alex Armstrong:

- **Proposal 1**
  - Creating the github repo and inviting the members
  - Software model choice
  - Requirements
- **Proposal 2**
  - Test plan for the software
  - Cost of hardware & software

Razvan Preotu

- **Proposal 1**
  - Software process model and requirements
  - Architectural design
- **Proposal 2**
  - Project scheduling
  - Estimated cost of personnel

Nga (Annie) Vu

- **Proposal 1**
  - Responding to instructor feedback to project proposal and plans to address this feedback
  - Sequence diagram
- **Proposal 2**
  - Estimated cost of personnel

Craig Hilby

- **Proposal 1**
  - Use Case Diagram
  - Architectural Design
- **Proposal 2**
  - Project scheduling
  - Cost, Effort, and Pricing estimation

Hareesh Parchuri

- **Proposal 2**
  - Project Scheduling
  - Comparison to similar existing software
  - References

Daniel Kiv

- **Proposal 1**
  - 1.4 (Making First Commit)
  - Class Diagram
- **Proposal 2**
  - Project scheduling
  - Cost, Effort, and Pricing estimation

Akhil Gangidi

- **Proposal 1**
  - 'Project_scope' commit
  - Making sure to put URL of team repository in project deliverable report
- **Proposal 2**
  - Test plan for the software

To recap the instructor feedback, our team must find and cite similar project implementations for deliverable 2. We need to add extra features to distinguish our project from already existing implementations and explain them. Finally, a comparison must be made between our design and the similar implementations. From initial research, there are a few similar open-source video-conferencing applications with a collaborative whiteboard. These include Apache OpenMeetings, BigBlueButton, and Mconf. Their features are audio/video conferencing, a moderating system, a chatroom, user and room management, and whiteboarding. There is an open-source educational whiteboard called OpenBoard which does not have text, video, or voice chats. What makes our design unique is that it is education-focused. Along with the features of the open-source video-conferencing applications, we have options to add student IDs, to post announcements, and to post and complete assignments. We plan to use a table to compare the features of our project and those other implementations.

1. Github URL

https://github.com/RazerNinjas/3354-RuntimeTerror

2. Task Delegation

Alex Armstrong:

- **Proposal 1**
  - Creating the github repo and inviting the members
  - Software model choice
  - Requirements
- **Proposal 2**
  - Test plan for the software
  - Cost of hardware & software

Razvan Preotu

- **Proposal 1**
  - Software process model and requirements
  - Architectural design
- **Proposal 2**
  - Project scheduling
  - Estimated cost of personnel

Nga (Annie) Vu

- **Proposal 1**
  - Responding to instructor feedback to project proposal and plans to address this feedback

    - Sequence diagram
- **Proposal 2**
    - Estimated cost of personnel

Craig Hilby

- **Proposal 1**
    - Use Case Diagram
    - Architectural Design
- **Proposal 2**
    - Project scheduling
    - Cost, Effort, and Pricing estimation

Hareesh Parchuri

- **Proposal 1**
    - Use Case Diagram
    - Sequence Diagram
- **Proposal 2**
    - Project Scheduling
    - Comparison to similar existing software
    - References

Daniel Kiv

- **Proposal 1**
    - 1.4 (Making First Commit)
    - Class Diagram
- **Proposal 2**
    - Project scheduling
    - Cost, Effort, and Pricing estimation

Akhil Gangidi

- **Proposal 1**
    - ‘Project_scope’ commit
    - Making sure to put URL of team repository in project deliverable report
    - Class Diagram
- **Proposal 2**
    - Test plan for the software

3. Software process model employed

Our group decided to use the prototyping model because our program is expected to add more features over a continued period of time. We prototype the program through a test build within the development location as well as a test build that users may opt into for the newest features in testing. Prototyping allows the program to add new features quickly rather than waiting for the next iteration to add completely giving users the best features as they are requested and to fix any bugs that may come up with the users.

4. Software requirements

Functional requirements:

- A user should be able to login to the server through a client
- A user should be able to send message to individuals or a group of users if they share the same class
- A user should be able to create a whiteboard
- A user should be able to modify a whiteboard if they have permission
- A user should be able to add,delete, and modify students in a class if they have permission

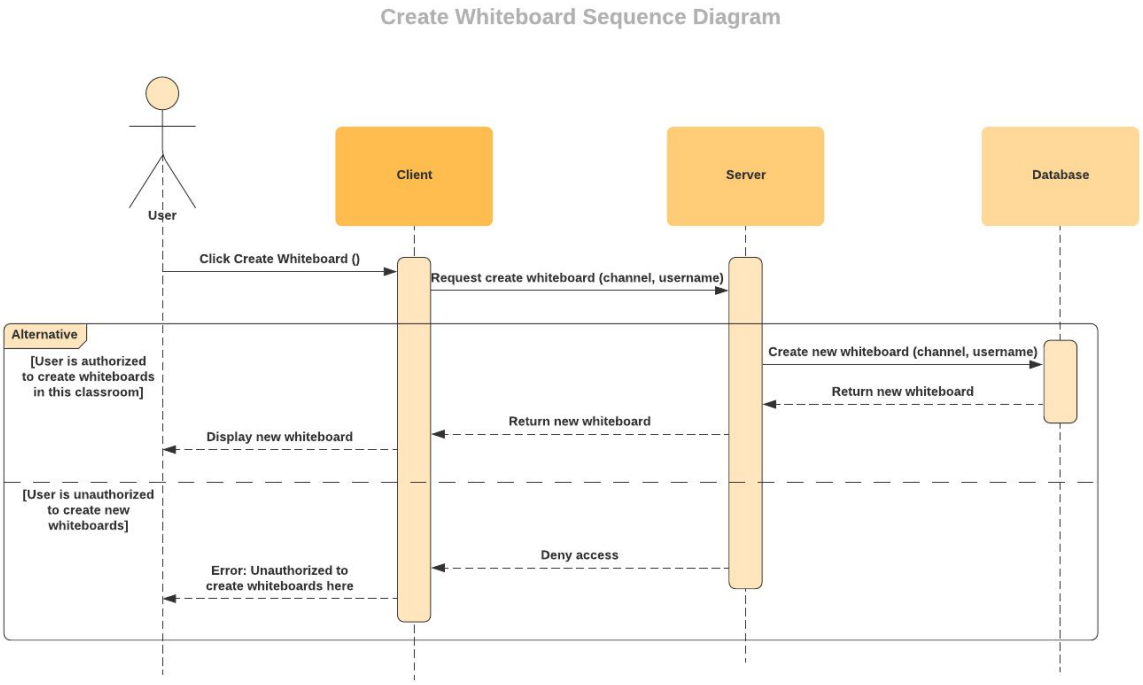
Non-functional requirements:

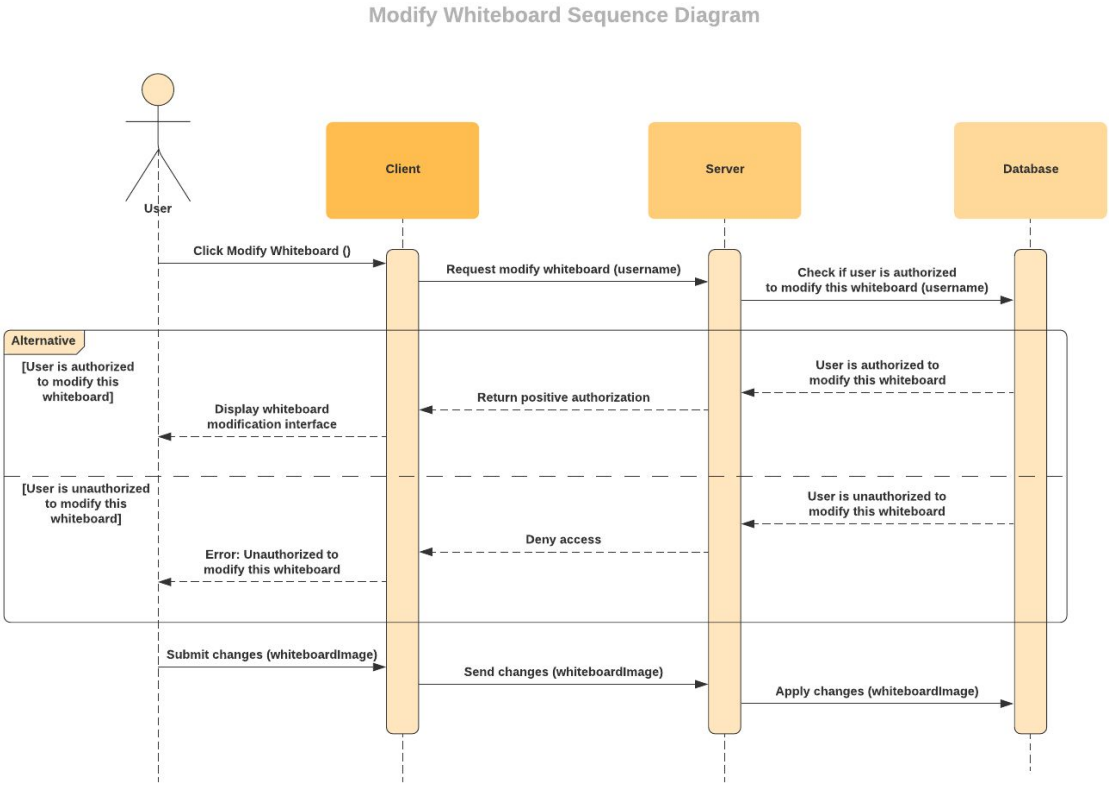
- Usability Requirements
    - A user should be able to understand how to navigate the program within 30 mins of use
- Performance Requirements
    - Logging in to the service should take 10 seconds maximum
- Space Requirements
    - A message maximum size is 1000 characters
    - The class maximum size is 100 students
- Dependability Requirements
    - The server and database should be up at all times for service
- Security Requirements
    - Users will sign up using their email and an 8+ character password
    - Implement 2-step verification
        - In case the user or someone else attempts to sign in on a different computer, a code will be sent to their email that is required to login
- Environmental Requirements
    - The server and database must be held on a server farm using AWS.
- Operational Requirements
    - The program is designed to be used for educational services private or public or personal (such as private tutoring).
- Development Requirements
    - The program must use a web based scripting language such as Python.
- Regulatory Requirements
    - The program shall be adjustable for various schools preferences in accordance to the education board.

- Ethical Requirements
    - The program shall implement anti cheating features to conform to academic integrity if students must send a document over the service.
- Accounting Requirements
    - The program should also be able to store payment records for accounting purposes.
- Safety Requirements
    - The program should implement anti cyberbullying features.

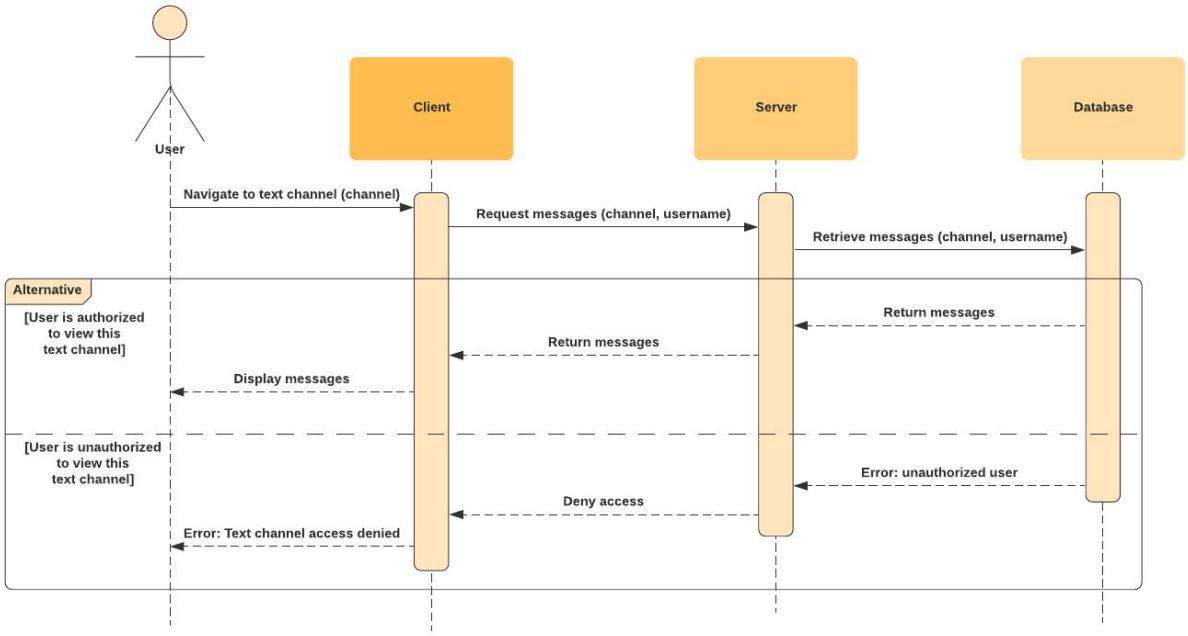
5. Use case diagram

6. Sequence Diagrams

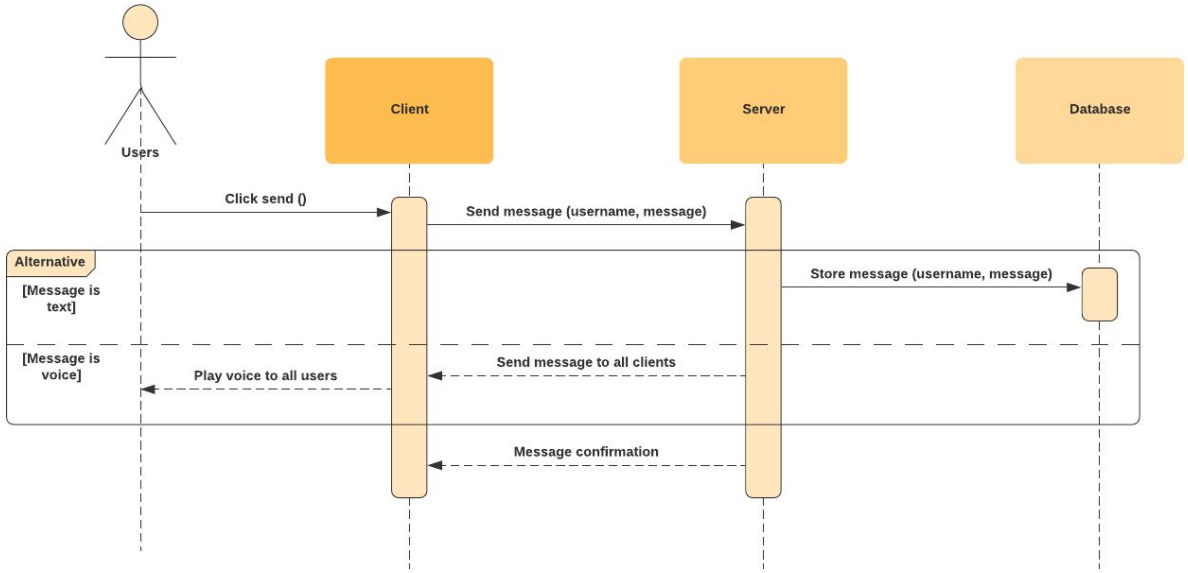




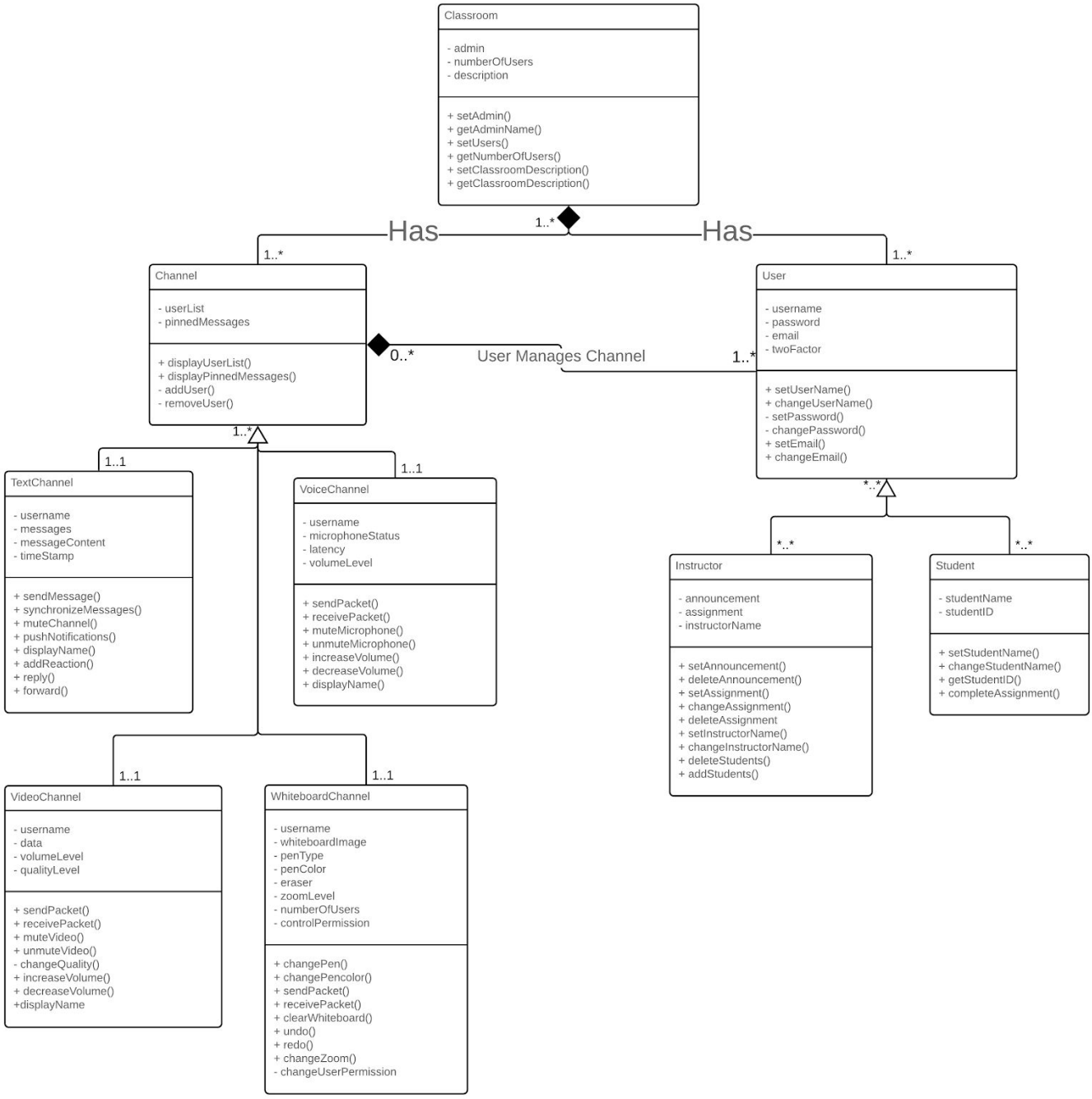
Recieve Text Messages Sequence Diagram



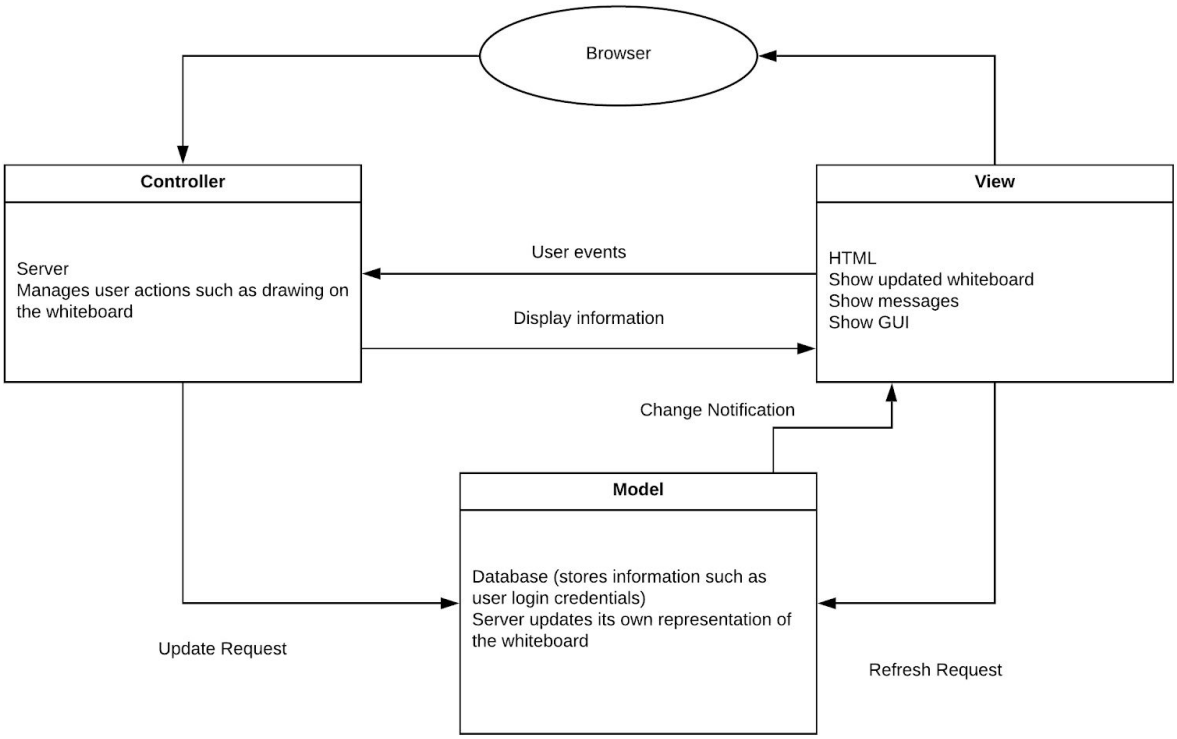
Send Voice/Text Message Sequence Diagram



7. Class Diagram



8. Architectural Design



Our group will be using the Model-View-Controller (MVC) architectural pattern. The model component will consist of a database to store the system data. It will be in charge of adding, retrieving, and modifying items from the database, such as user login credentials. The view component will define and manage how information is displayed to the user. Because our project is a web application, HTML is a good option to use. The controller will consist of a server that will handle all user interactions by communicating with the model and view components. When the user interacts with the view, the server will communicate with the database to retrieve the necessary information. The server then communicates with the view to display the new information.

MVC is the most common architectural pattern for web-based applications because of its many advantages.

- Supports parallel development, as developers can be working on different components at the same time. This makes the project easier and quicker to implement.
- Allows the developers to create multiple views for a model.
- Modification of one component will not affect the entire system.
- Makes it easy to develop Search Engine Optimization friendly URLs which increases online traffic [1]

References

[1] Interserver Tips. (2019). *What is MVC? Advantages and Disadvantages of MVC - Interserver Tips*. [online] Available at: https://www.interserver.net/tips/kb/mvc-advantages-disadvantages-mvc/ [Accessed 17 Oct. 2019].