

CS3354 Software Engineering
Final Project Deliverable 1

Virtual Reality Staging and Touring

Group 4:

Dylan Stimac | Zachary Gray | Hafiz Akinteye | James Baker | Jeremy Culver | Matthew Tate |
Nasya Capetillo

1.

Project Goal:

Design and implement virtual reality staging and touring software that can be used in the real estate and tourism sectors.

Project Motivation:

Virtual reality software is still in its infancy and has extreme growth and implementation potential.

With staging and touring software, tourism can take place anytime, anywhere. Museums, landmarks, and other attractions could be uploaded to VR touring databases and allow users to view these attractions at different points in history from the comfort of their homes.

Further, the real estate sector suffers from issues related to potential buyers being unable to tour homes that they may have an interest in living in. Currently, buyers must rely on scheduling home tours with a real estate agent and balancing the schedules of multiple parties can prove very difficult. Another common issue occurs when buyers need to move out of state/country and are unable to truly experience a future home. Virtual touring will allow users to truly experience and fall in love with their options before committing to one of the largest investments of their lives.

Task Delegation:

- **Person in charge of submitting documents:** Hafiz

1st Deliverable:

The following will list the group members and their assigned tasks or areas of focus.

- **Dylan** - In charge of outlining the software process model used for the project.
- **Zach** - Handling the software requirements including both functional and non-functional requirements.
- **James** - Creating the use case diagrams. Revising documents before submission.
- **Hafiz** - Designing the class diagram including all classes and the relationships between them.
- **Jeremy** - Outlining the classes that will be used for the project
- **Nasya** - Choosing and applying an architectural design for the project.
- **Matthew** - Creating sequence diagrams for the use cases of the project.

2nd Deliverable:

The following will list the group members and their assigned tasks or areas of focus.

- **Dylan** - Cost estimations, effort invested, pricing and any other financial planning involved in the project
- **Zach** - Project scheduling, along with any visual aids that may be used to help plan or prepare for that including charts, calendars displaying tasks, ect.
- **James** - Software Test Planning and Results. Revising documents before submission.
- **Hafiz** - Research and compare products similar in design to our current project. Provide citations of sources to support it with the use of IEEE referencing.
- **Jeremy** - Summarize problems/flaws with the project
- **Nasya** - Conclusion statement expanding on any changes that arose, justifications for changes, and final reflection /evaluation of work.
- **Matthew** - Slide design, layout, and presentation planning.

Final Project Proposal

Good proposal and fair distribution of tasks to group members.

In your final project report (deliverable 2) please make sure to include the following:

- A thorough search to find similar application implementations. Please cite these work using IEEE citation format provided on Final Project Specifications document.
- Please make sure to differentiate your design from existing similar applications by including extra features into it.
- Please make sure to explicitly specify those differences by comparing your design with those existing similar applications.

In response to the feedback we received on our Final Project Proposal for the second deliverable we make sure to have searched for any similar application implementations and cite them by using the IEEE format that you have provided and as suggested. For the time being we will assign that to Hafiz, as that will fall under the same material and task he was originally delegated in the second deliverable.

Once the research has been accomplished we will include unique features to our project by either creating a comparison chart of other products on the market, or dedicating a specific slide in our presentation as to why our product is unique in comparison to any other software out on the market currently.

Lastly, after doing the research if we find that there are many software products on the market out there that are very similar to the one we are creating then we will brainstorm together some ideas so that we may provide extra services that make our project even more unique and include that in our presentation.

2. **OUR REPOSITORY URL LINK: https://github.com/Nasya-C/3354_Group4/ ***

3. Delegation of tasks:

Dylan Stimac:	Outlining Software Process Model
Nasya Capetillo:	Project Leader, Completed Github steps: 1.1 - 1.3, Applying Architectural
Design Model:	Filling out this Delegation of Tasks, Final Project Draft Description
Hafiz Akinteye:	Document Submission, Designing Class Diagrams,
Jeremy Culver:	<i>Outlining the classes that will be used for the project *needs confirmation*</i>
Matthew Tate:	Creating Sequence Diagrams for Use Cases
Zachary Gray:	Handling Software Requirements both functional and non-functional.
James Baker:	GitHub step 1.4, Creating Use Case Diagrams, Document Revision

4. Software Process Model

Incremental Process Model

The incremental model fits perfectly with our MVC software design pattern. The incremental model allows for the product to be built module by module, which is essential. It is vital that the virtual reality staging and touring software becomes functional, piece-by-piece, until it reaches final stages of development. Prior to any possibility of users being able to virtually tour homes, a software module must be developed that is able to accurately capture virtual models of homes that is compatible with modern virtual capture/staging cameras. Once reality capture is functional in the software, the HUD, Heads-Up-Display, would need to be developed and implemented. The HUD provides pertinent, toggleable information about the property such as: sqft. of each room, number of rooms, gas/electric hookups, survey lines, etc. that users would need in making the decision to further pursue the purchase of the real estate property. Next, the ability to edit and update property information is required. The goal of the software is to provide accurate, up-to-date information about each property, so that potential buyers can trust that pursuing a property, based on their virtual tour, is sound. In order to ensure true representations of the real estate properties is available to users, the ability to access and edit/update the database storing all virtually tourable properties is necessary. Finally, user-access and login to the servers storing the virtually tourable properties completes the software. For real estate companies to employ our software, the ability for users to create accounts, tour properties, and favorite properties is pertinent to the usability of the virtual reality staging and touring software. Since the software process follows a modular, linear process, the incremental process model is undoubtedly the correct choice for the software design and implementation.

5. Software Requirements

5.a.) Functional Requirements:

Users must be able to virtually tour a wide variety of homes they are interested in buying.

Users must also be able to browse locations to tour and have the ability to add any locations to a favorite list.

Sellers must have some functionality for them to upload videos/renders of the home they are selling.

Agents must be able to communicate with users while the virtual tour takes place to answer questions.

5.b.) Non-functional requirements

Usability: The product must be easy to use for consumers, and easy to use for people uploading their home tour

Performance: The program should run at least 30fps, and should require content submitted by sellers to be of high quality

Space: The program should not take up an excessive amount of storage space

Dependability: The program should run correctly every time unless it is down for routine maintenance

Security: Only authorized users should be able to log in and view homes for sale. Only approved and screened sellers should be able to upload their homes for sale

Environmental: This project should not have any environmental impact but will have to comply

with United States law regardless

Operational: The finished product must be able to be used from every standard VR device, and have easy to use signups for buyers and sellers

Development: This project will be developed by our team in a timely fashion, implemented using the MVC pattern

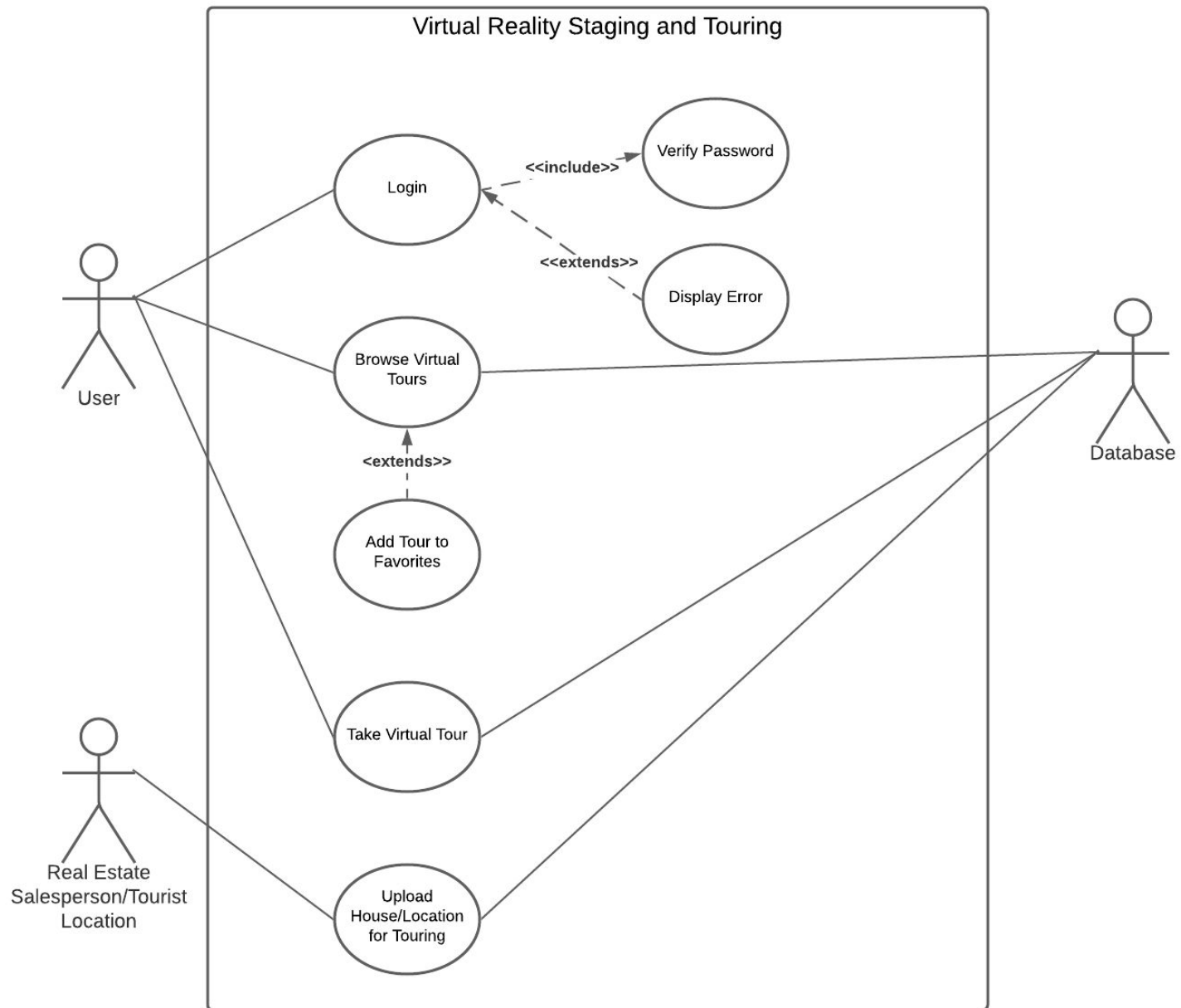
Regulatory: This project will need to comply with consumer privacy laws in the United States and during its development must abide by workplace regulations

Ethical: This project must allow users to remove their homes from the site if they do not feel safe, and protect users from abuse of the system by unauthorized access

Accounting: This project must keep an accurate record of all funding received, and track expenses to give shareholders their fair share of returns

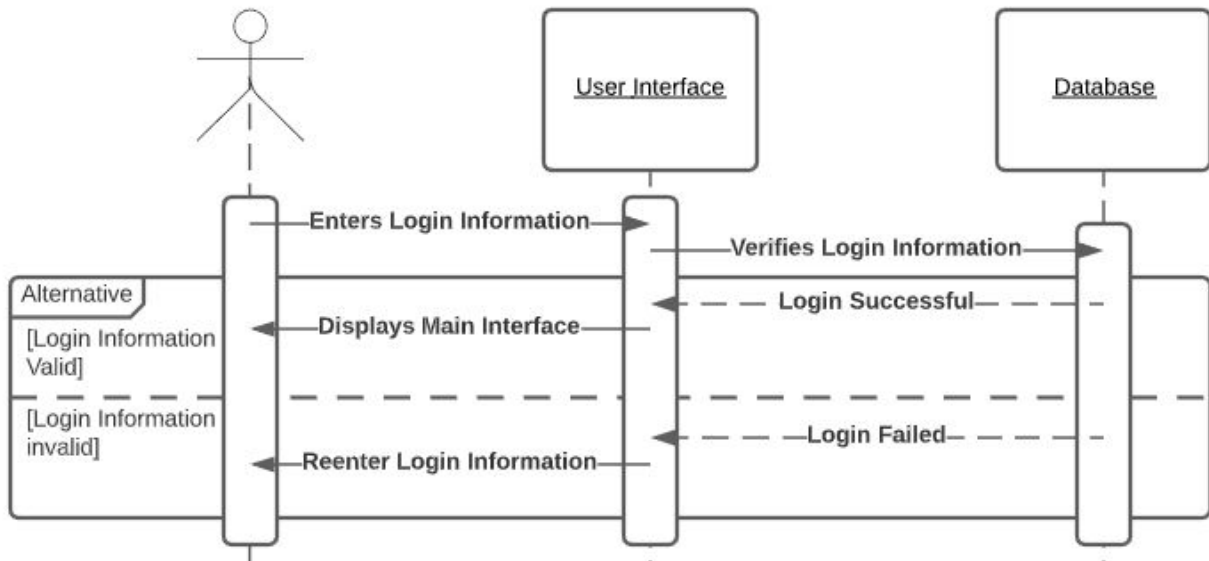
Safety/Security: This project needs to abide by US law regarding software security, and have adequate protections to make sure no unauthorized access occurs, and that authorized users do not abuse the power they have.

6. Use Case Diagram

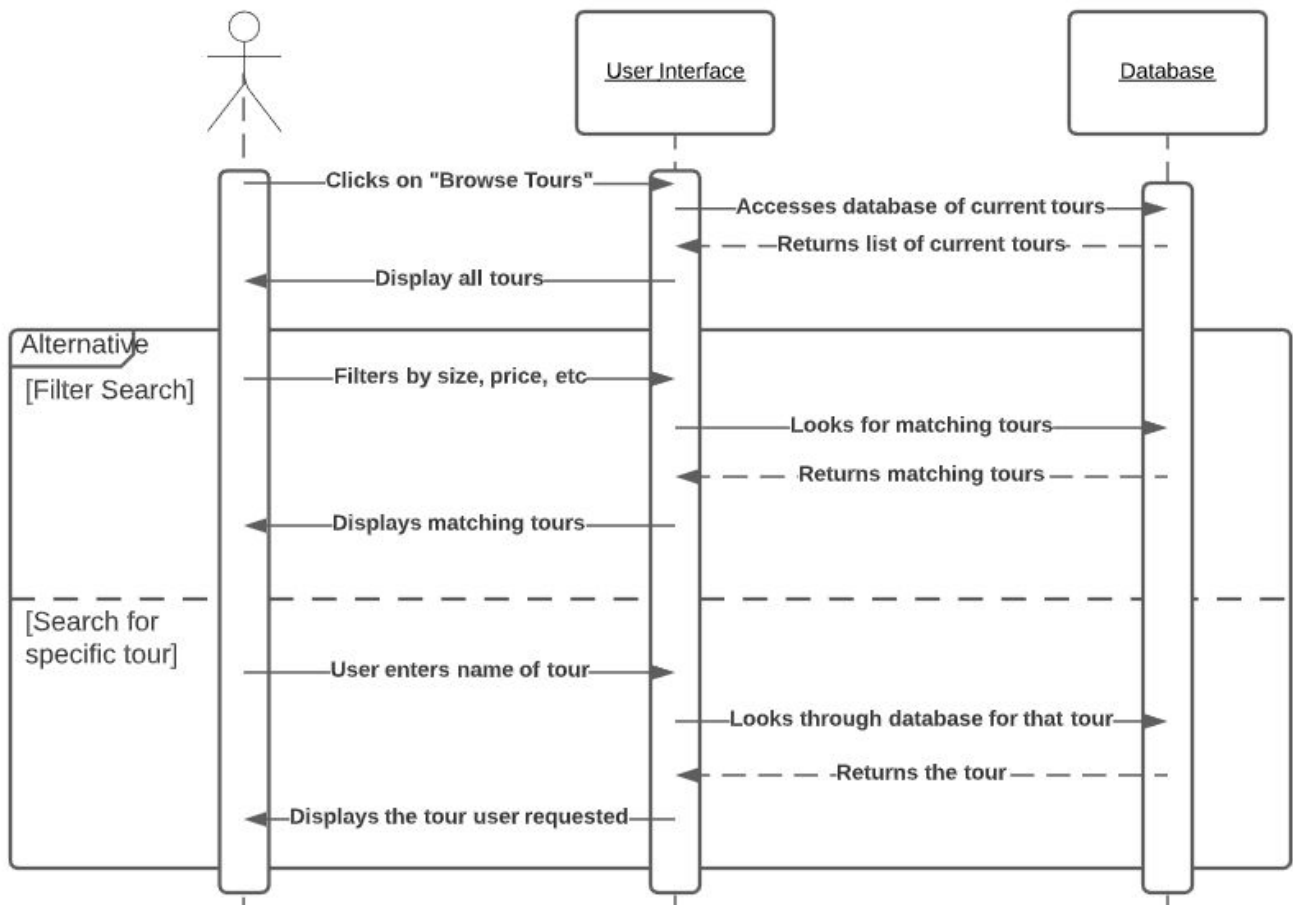


7. Sequence diagram

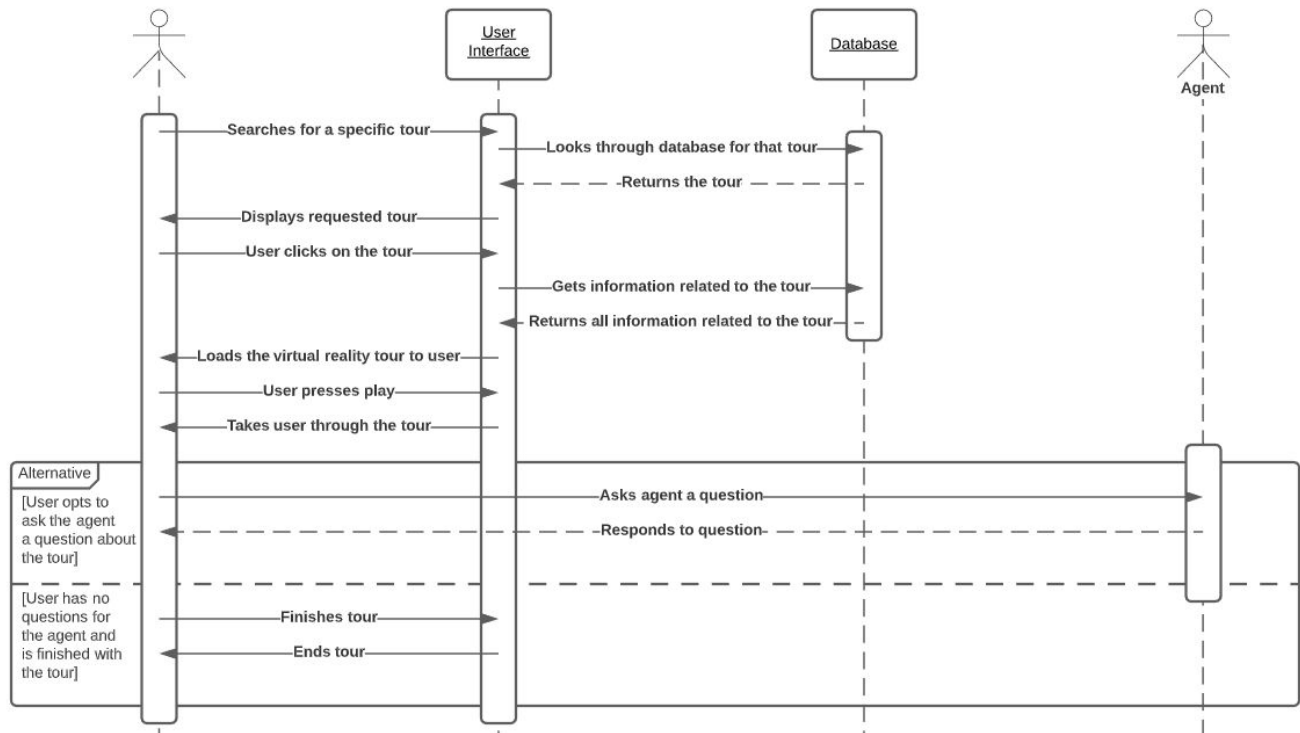
Login Sequence Diagram:



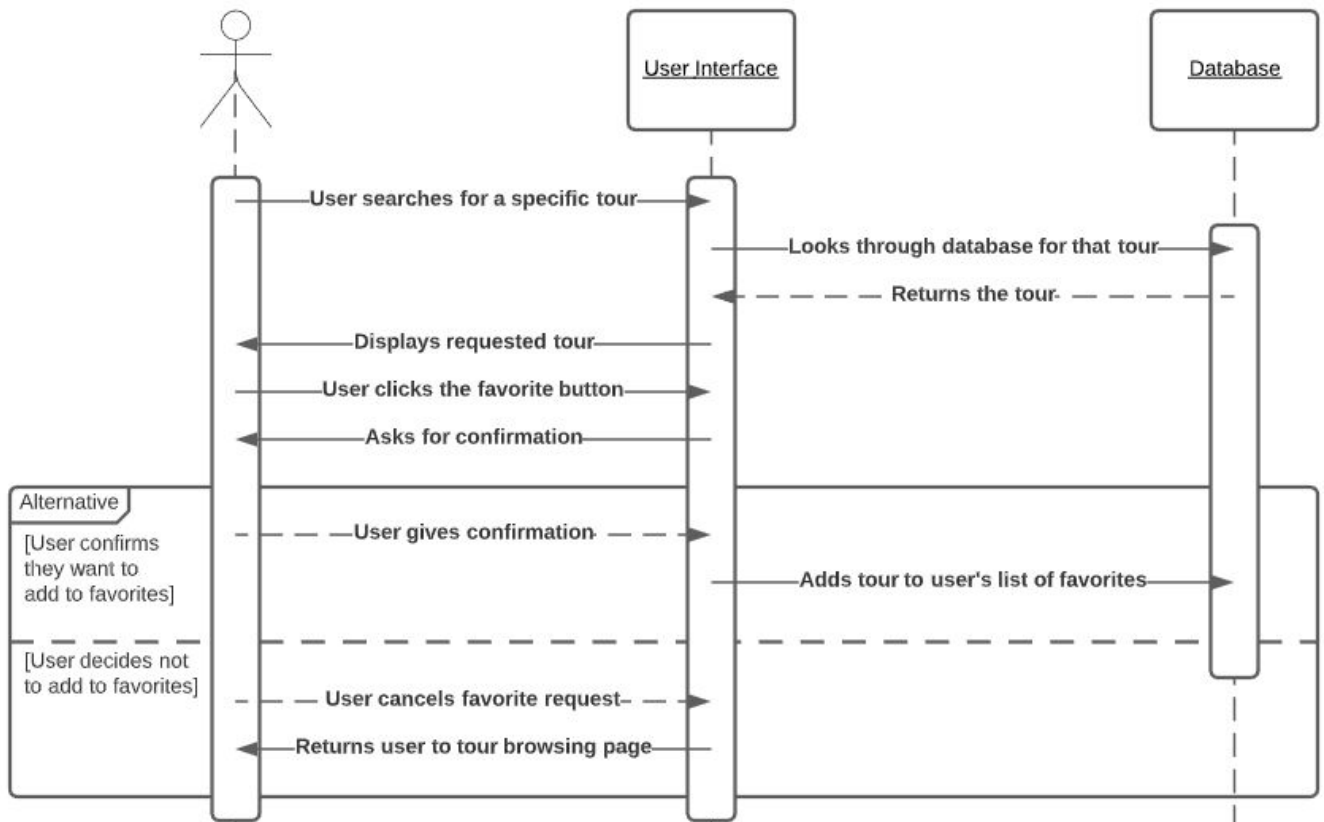
Browsing Tours Sequence Diagram:



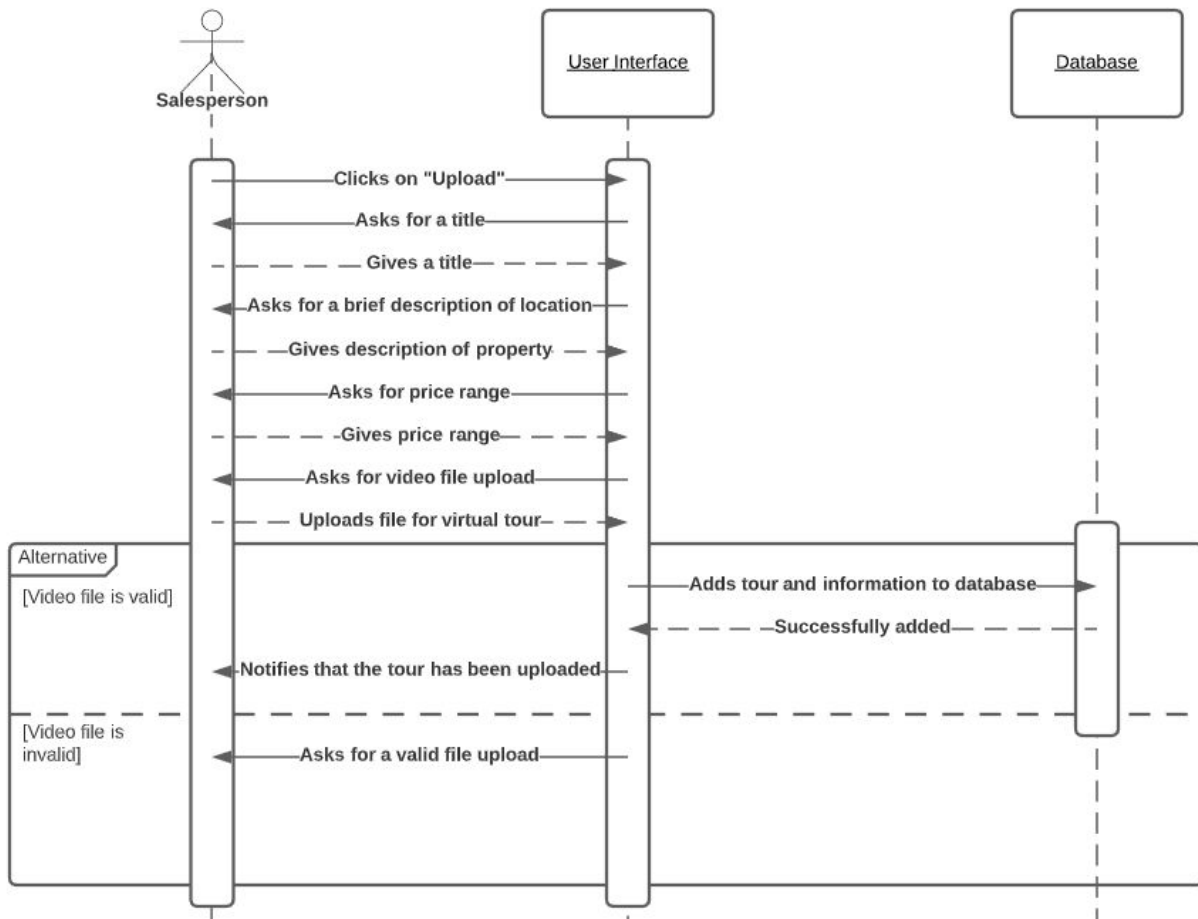
Take Virtual Tour Sequence Diagram:



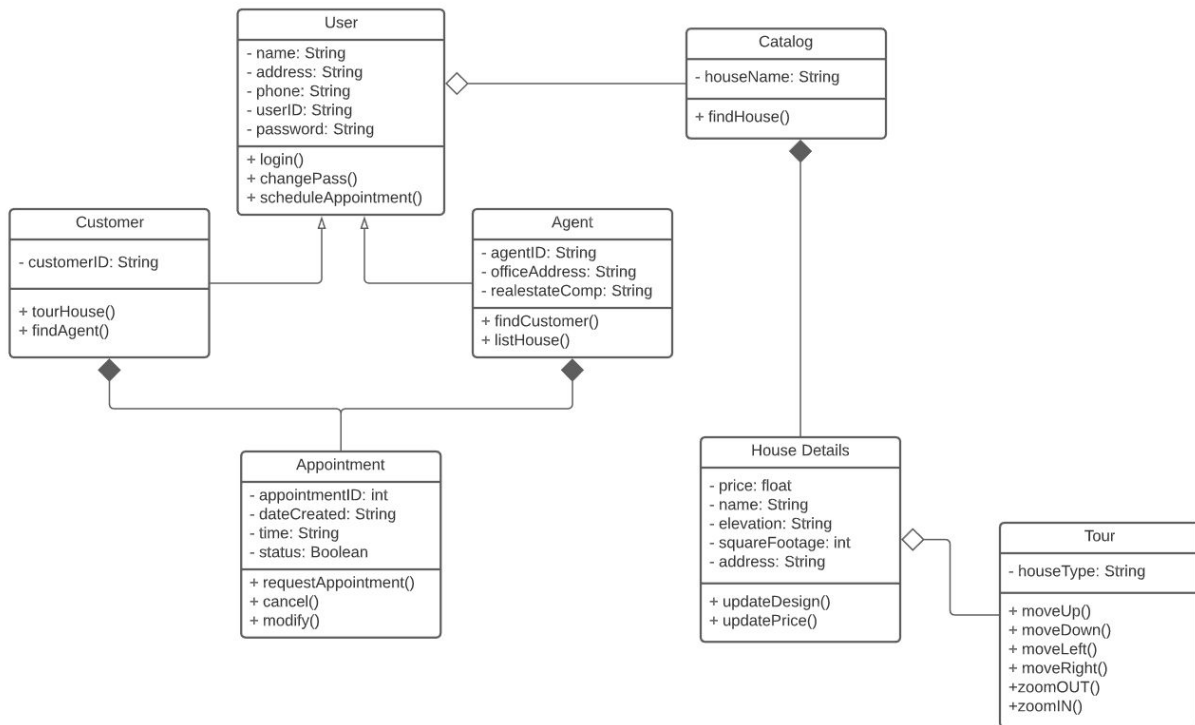
Adding Tour to Favorites Sequence Diagram:



Uploading House/Location for Touring Sequence Diagram:

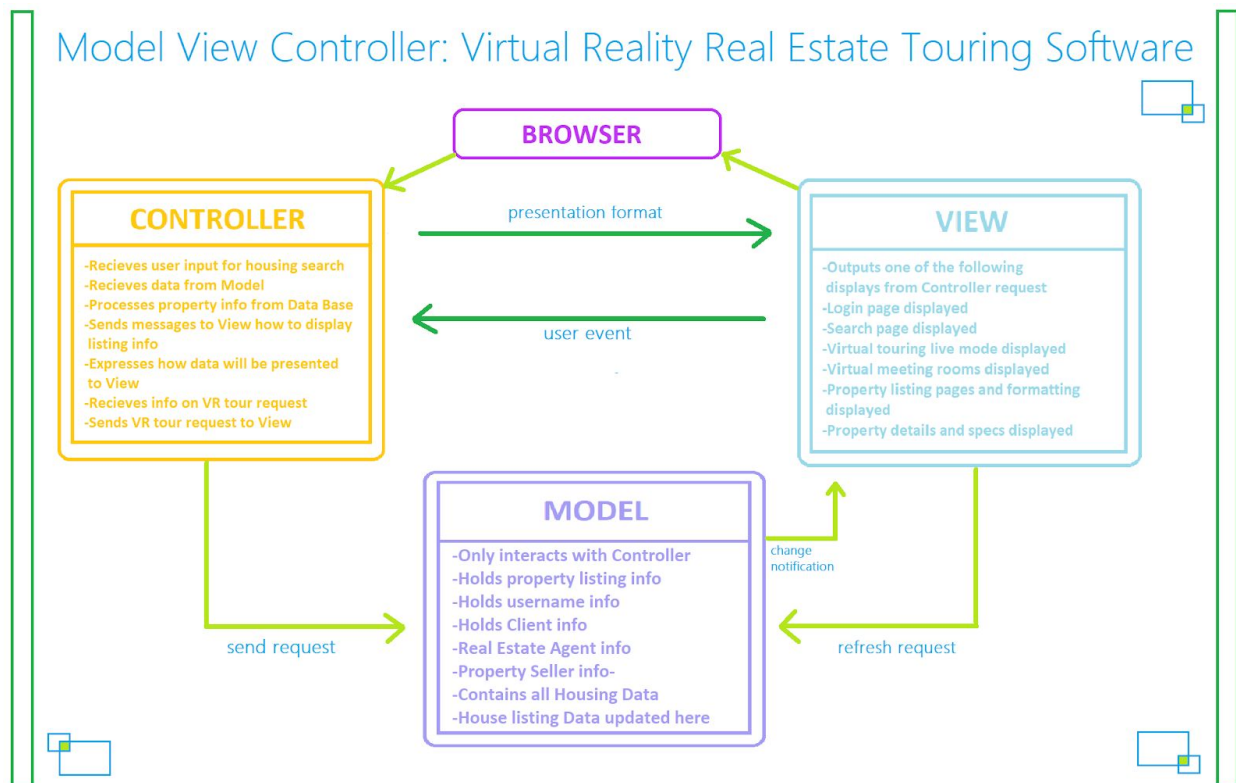


8. Class diagram



9. Architectural design

9.1. Model-View-Controller (MVC) pattern



The main architecture pattern we chose was the MVC pattern for our group. The reasoning behind our choice is due to the clear separation of the systems data in the software from the visual display of the data. This is beneficial in the sense that the software will be used to present the same data about multiple house listings to the clients, in multiple ways visually to suit their ever changing preferences or needs. Furthermore, it will also be helpful because as housing listings in the real estate market are constantly changing, we will also be able to update the data as often as needed without having to make changes to the whole system's code or program as a whole. The other alternative our group seemed to like was client server as the client's could log on anywhere in the world, since the client's will be most likely in various locations when looking at the potential homes, since that directly relates to the niche of the software we are creating is trying to meet but since the MVC model is capable of that too if we do it per say on a webpage or mobile app, then people can also still log on from anywhere. Furthermore the other reason they considered client server pattern was because they felt that the client server pattern type was less limiting in its structure and it provided some freedom. But like I previously mentioned, all in all for now our group settled on the MVC model as it seems to have stronger reasonings behind its structure. Lastly it seems like it also has a superior security structure compared to the client server pattern, as it can not be attacked as easily as client server can since client server pattern is more susceptible to server attacks.