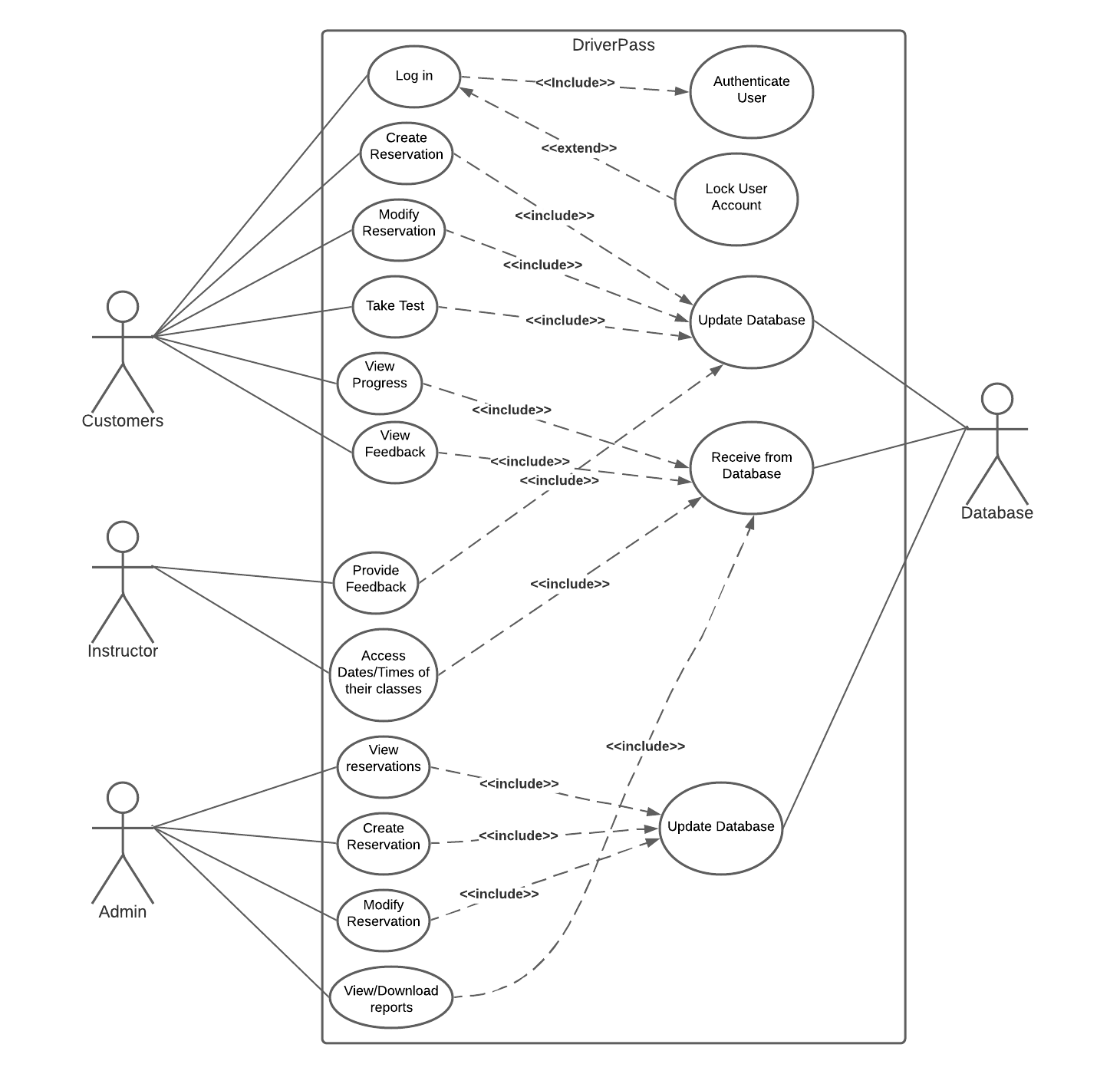
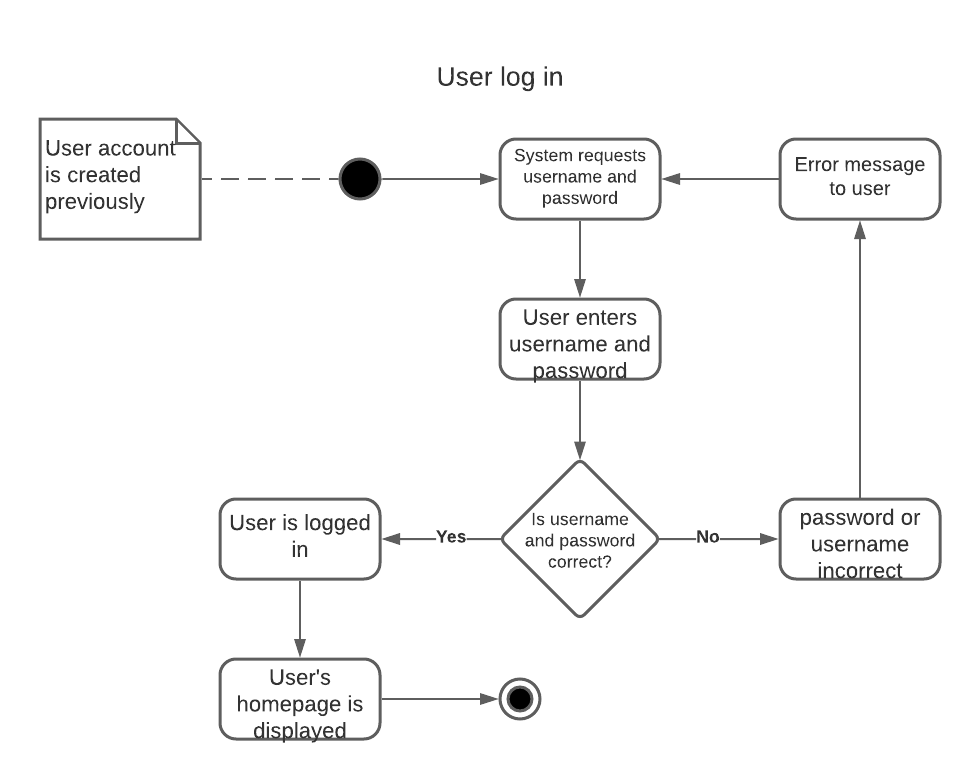
# CS 255 System Design Document Template

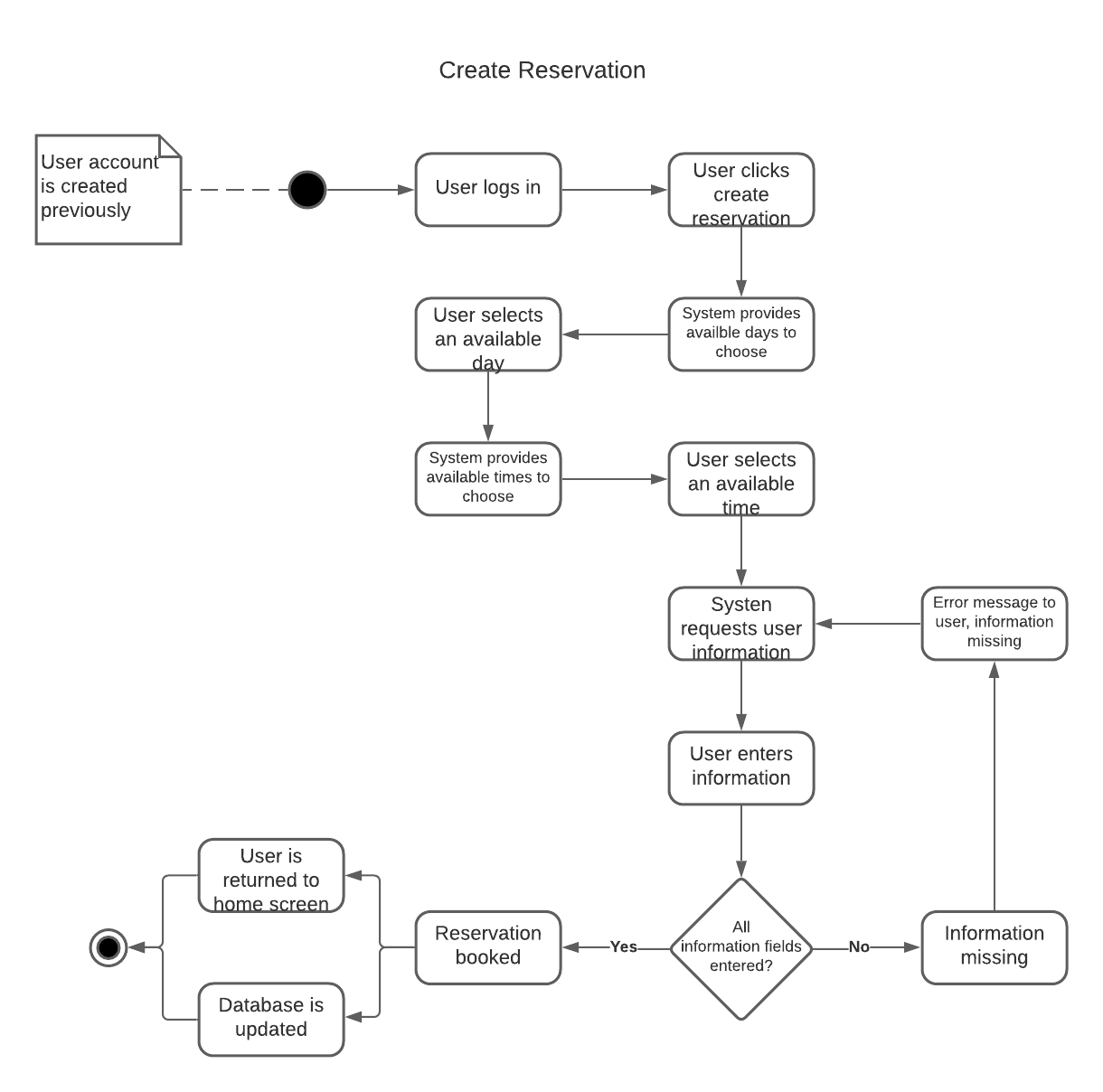
## UML Diagrams

### UML Use Case Diagram

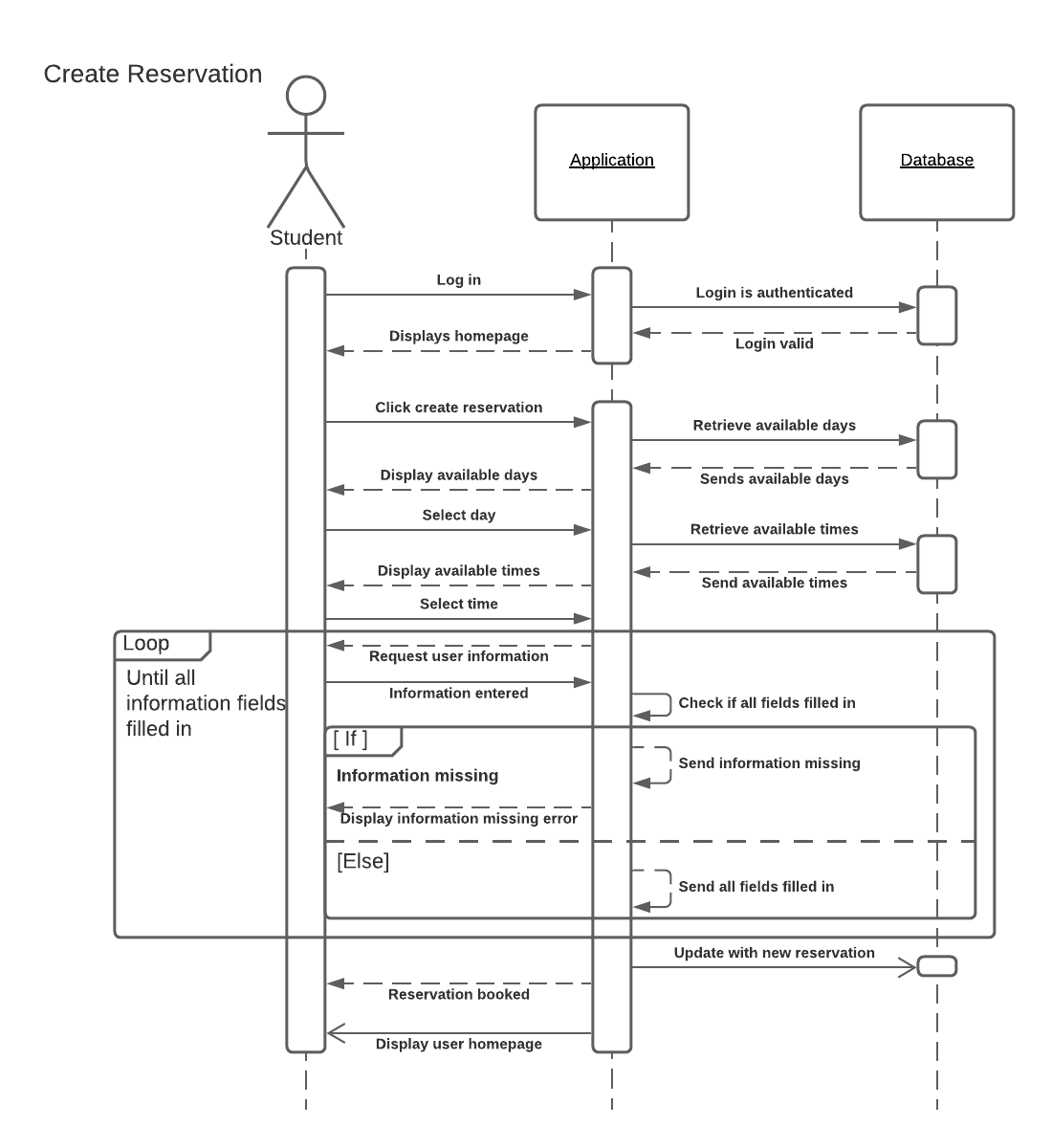


### UML Activity Diagrams

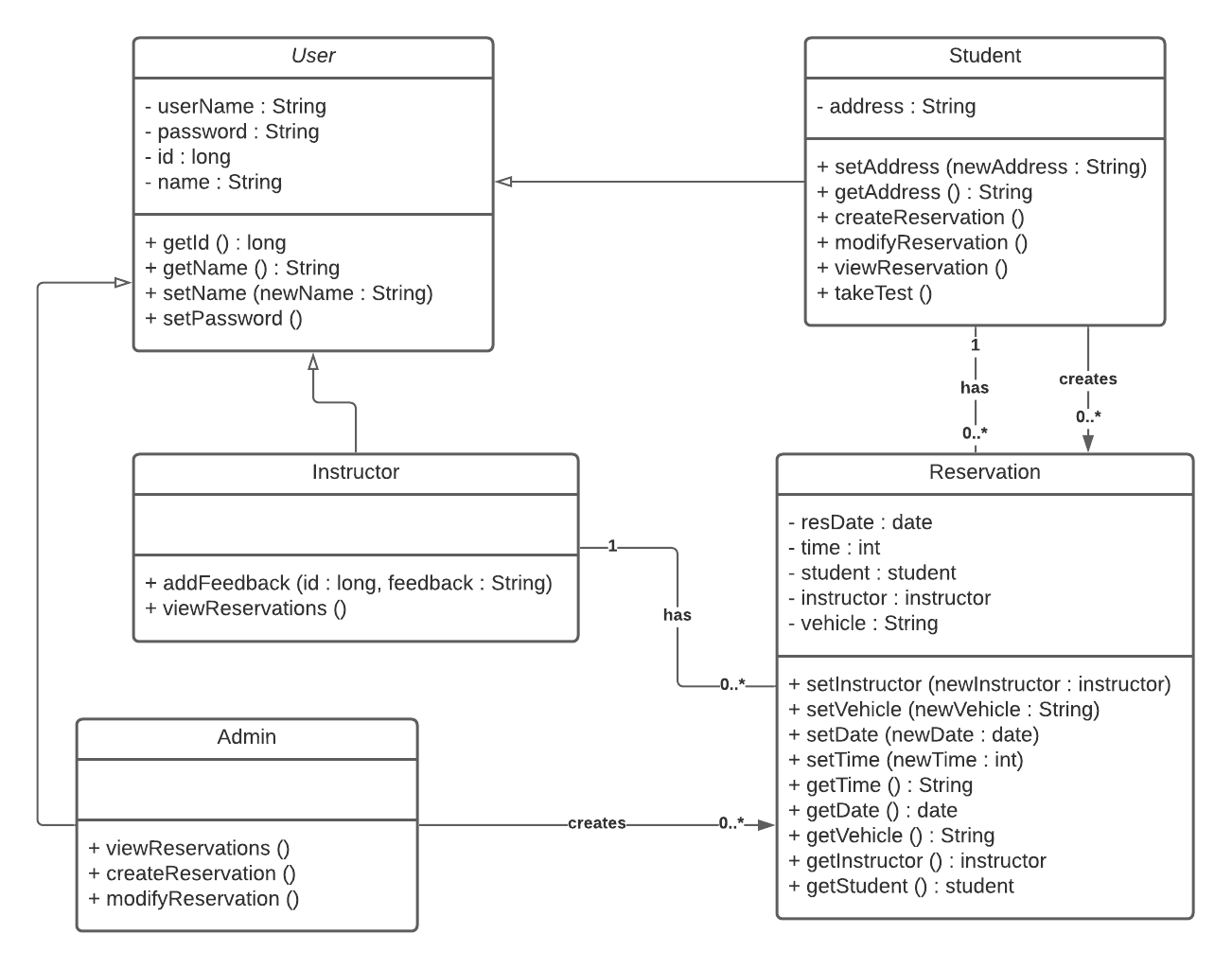




### UML Sequence Diagram



### UML Class Diagram



## Technical Requirements

*[Based on the diagrams you have created, describe the technical requirements of your system. These requirements should address the required hardware, software, tools, and infrastructure necessary for your system design.]*

* Requires a database
* Requires an interface
* Requires a server to host
* Will need the performance to adapt to multiple users
* Needs to be reliable
* Requires access to the internet
* Needs to be web-based
* The system should run without significant time between interactions
* The system should not slow down as the number of users increases
* The system needs to allow multiple operating systems to connect to the web-based system
* A database will be required to store customer information
* An interface will be used for the users to interact with the system
* **Operating Platform**: A Linux based platform will be cheap, reliable, and secure
* **Storage Management**: RAID 5 will offer redundancy and still perform well with keeping in mind of budget. This requires less hard drives than RAID 1.
* **Distributed Systems and Networks**:
  + Using distributed systems, especially a three-tier system will allow access from various platforms. The three-tier system is a distributed system that has three layers the presentation (client) layer, the application layer, and the database layer. The presentation layer is what the user interacts with. This will be accessed through the user’s browser that connects to a web server. The user interacts with the presentation layer which in turn passes information to the application layer. The application layer is usually on a different server and processes the information that is passed to it from the presentation layer. Then the application layer communicates with the database layer in order to accomplish the user’s requests or inputs. The database layer then executes the actions needed and sends information back to the application layer. The application layer processes this information and sends it back to the presentation layer. This will allow different browsers and ultimately different platforms to access the webserver and interact with the application.
  + Each layer is usually on a different server, this allows for optimum performance for each layer. Each layer will have to connect to the next layer to communicate to it which can be an issue if a connection fails or can be a bottle neck that slows things down.
  + Using the cloud to host the layers can help with performance and reliability. Using the cloud also offers easy scalability.
* In order to support multiple browsers, it would be best to use things that are compatible on multiple browsers. Expertise within HTML is important, having the knowledge of cross-browser compatible libraries and frameworks is important to know. Having expertise in CSS is also important in order to understand how to reduce browser inconsistencies across multiple browsers. As an example, using the Foundation framework would help to ensure cross-browser compatibility.
* **Programing Languages** – HTML, CSS, and Javascript
* **IDE** – Some examples include: Visual Studio, RJ TextEd, and Light Table
* **Javascript** **libraries**: examples include jQuery, Dojo, and React
* **CSS** **preprocessor:** like Sass
* **Frameworks:** like Foundation and Bootstrap