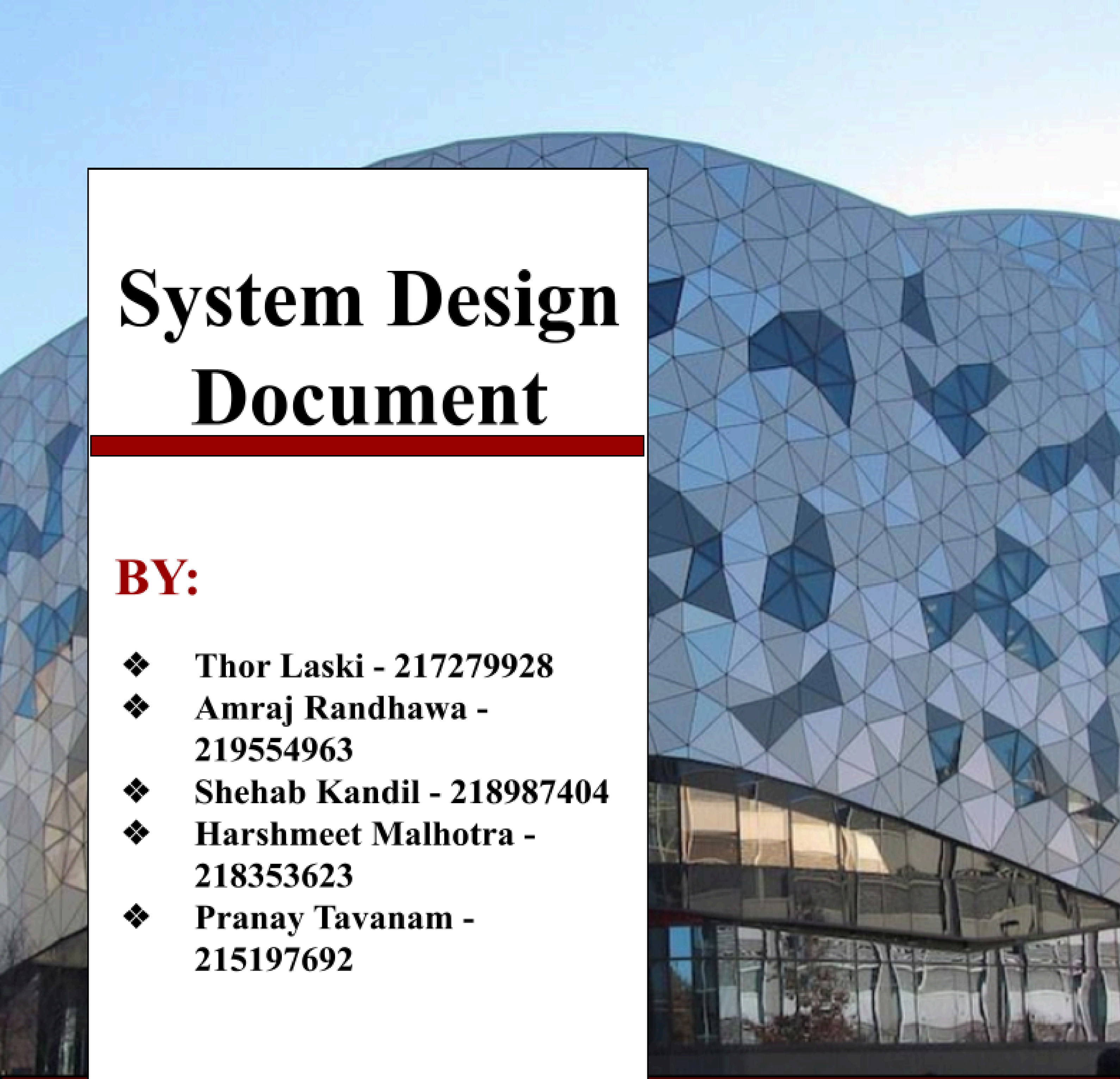


System Design Document

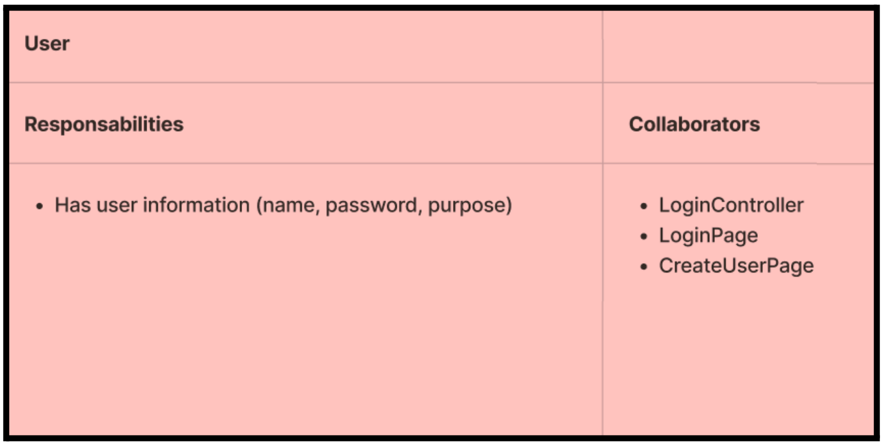
BY:

- Thor Laski - 217279928
- Amraj Randhawa - 219554963
- Shehab Kandil - 218987404
- Harshmeet Malhotra - 218353623
- Pranay Tavanam - 215197692

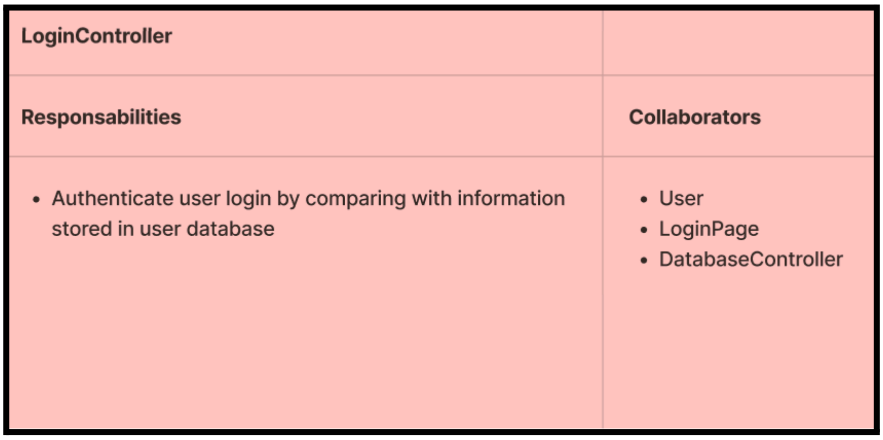
November 3
2025

Table of Content

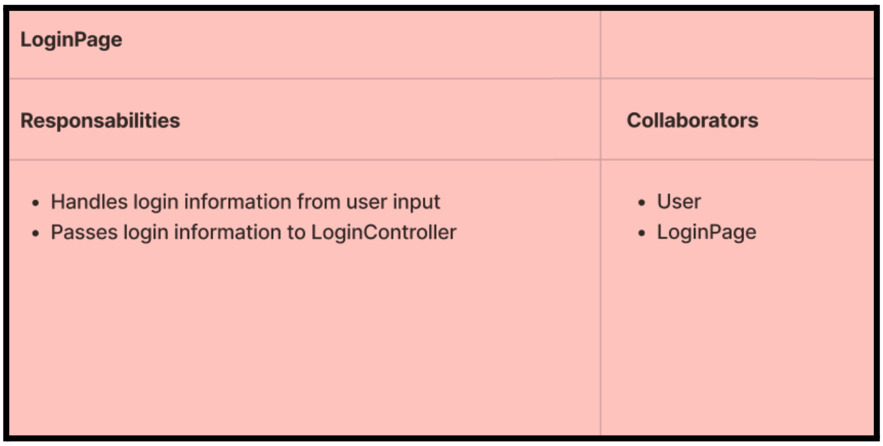
CRC High Level Class Description



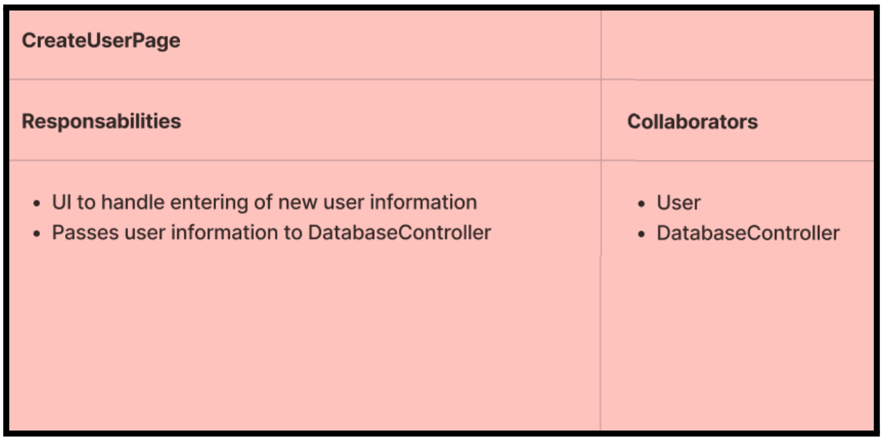
User	
Responsabilities	**Collaborators**
- Has user information (name, password, purpose)	- LoginController - LoginPage - CreateUserPage



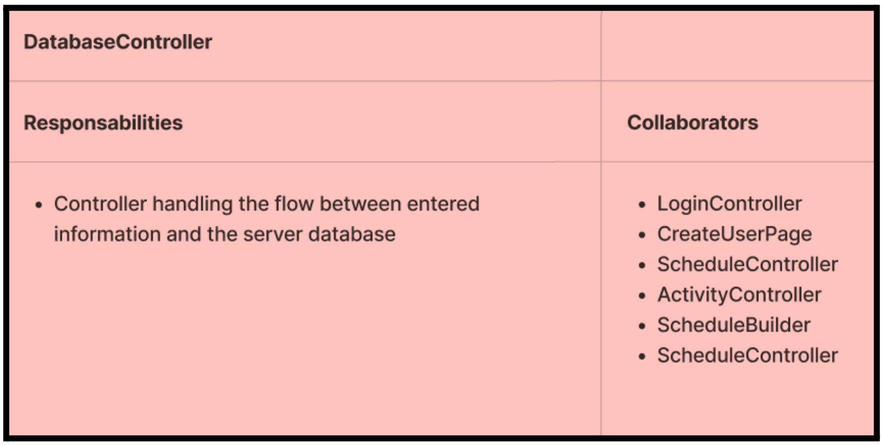
LoginController	
Responsabilities	**Collaborators**
- Authenticate user login by comparing with information stored in user database	- User - LoginPage - DatabaseController



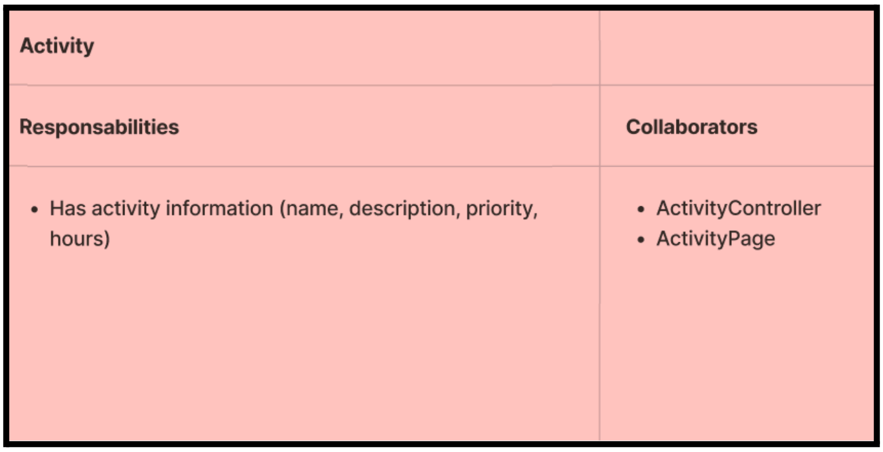
LoginPage	
Responsabilities	**Collaborators**
- Handles login information from user input - Passes login information to LoginController	- User - LoginPage



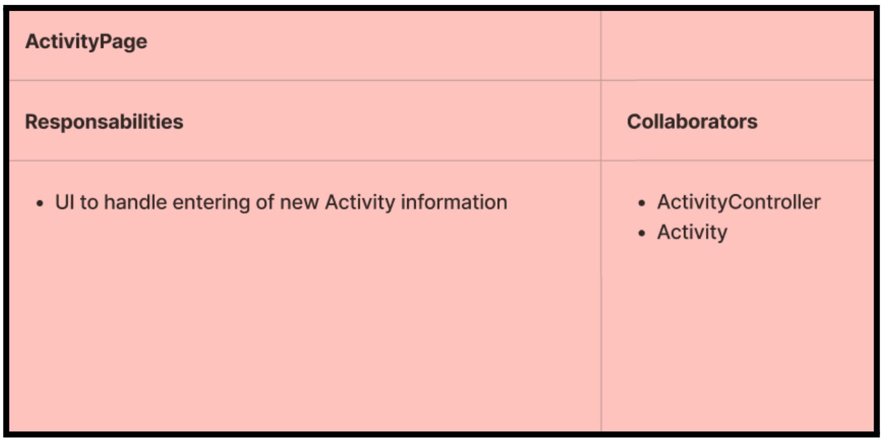
CreateUserPage	
Responsabilities	**Collaborators**
• UI to handle entering of new user information • Passes user information to DatabaseController	• User • DatabaseController



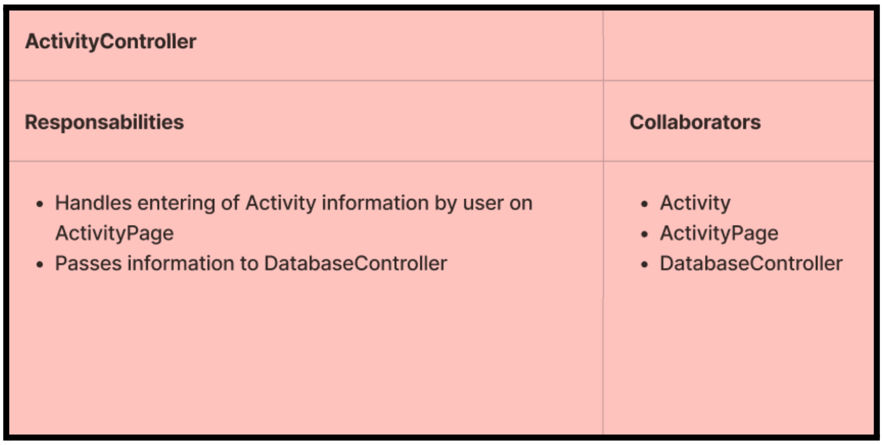
DatabaseController	
Responsabilities	**Collaborators**
• Controller handling the flow between entered information and the server database	• LoginController • CreateUserPage • ScheduleController • ActivityController • ScheduleBuilder • ScheduleController



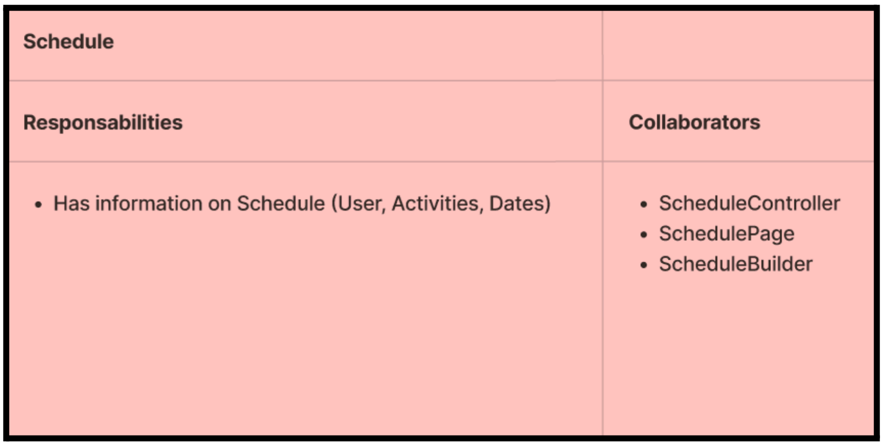
Activity	
Responsabilities	**Collaborators**
• Has activity information (name, description, priority, hours)	• ActivityController • ActivityPage



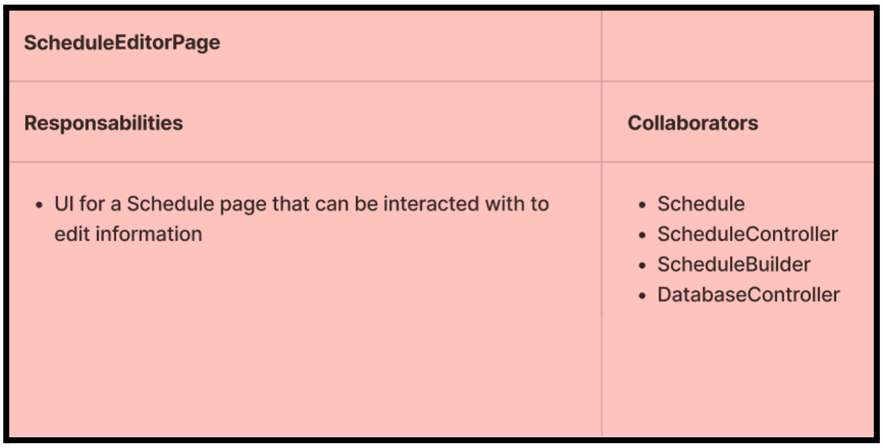
ActivityPage	
Responsabilities	**Collaborators**
• UI to handle entering of new Activity information	• ActivityController • Activity



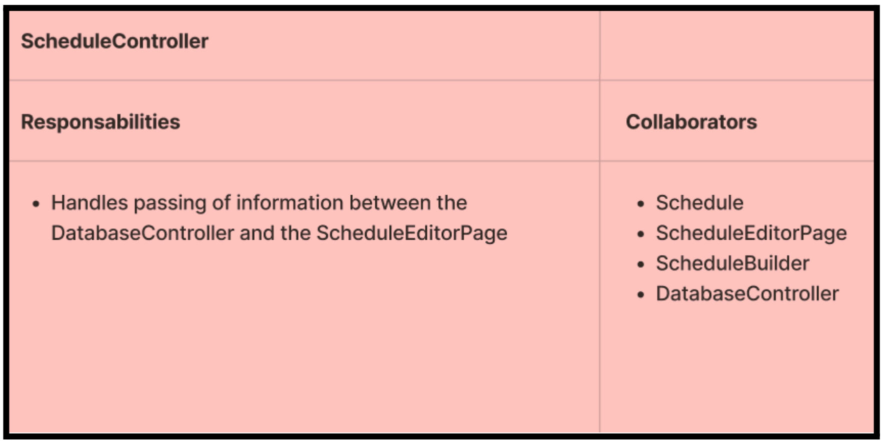
ActivityController	
Responsabilities	**Collaborators**
• Handles entering of Activity information by user on ActivityPage • Passes information to DatabaseController	• Activity • ActivityPage • DatabaseController



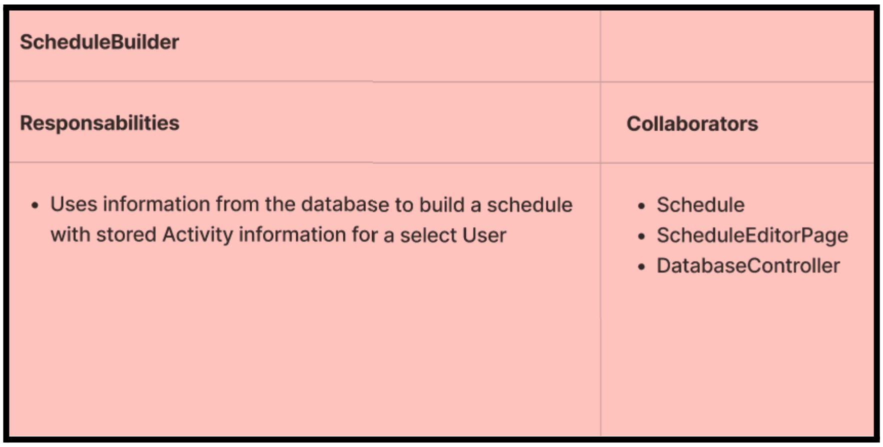
Schedule	
Responsabilities	**Collaborators**
• Has information on Schedule (User, Activities, Dates)	• ScheduleController • SchedulePage • ScheduleBuilder



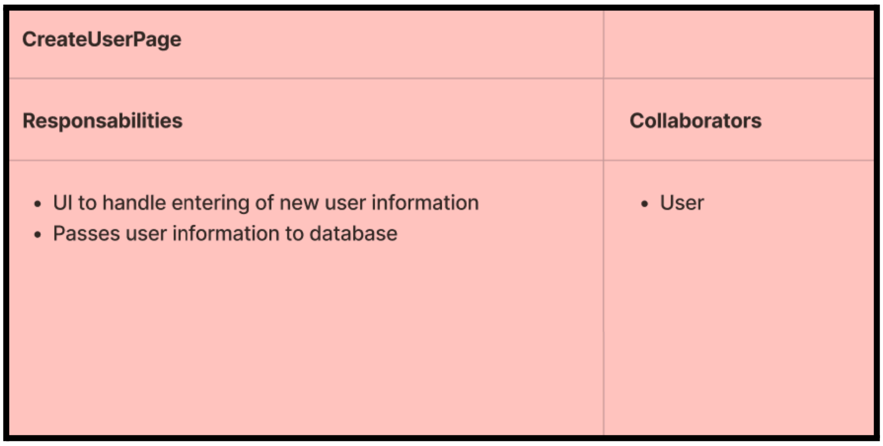
ScheduleEditorPage	
Responsabilities	**Collaborators**
• UI for a Schedule page that can be interacted with to edit information	• Schedule • ScheduleController • ScheduleBuilder • DatabaseController



ScheduleController	
Responsabilities	**Collaborators**
• Handles passing of information between the DatabaseController and the ScheduleEditorPage	• Schedule • ScheduleEditorPage • ScheduleBuilder • DatabaseController



ScheduleBuilder	
Responsabilities	**Collaborators**
• Uses information from the database to build a schedule with stored Activity information for a select User	• Schedule • ScheduleEditorPage • DatabaseController

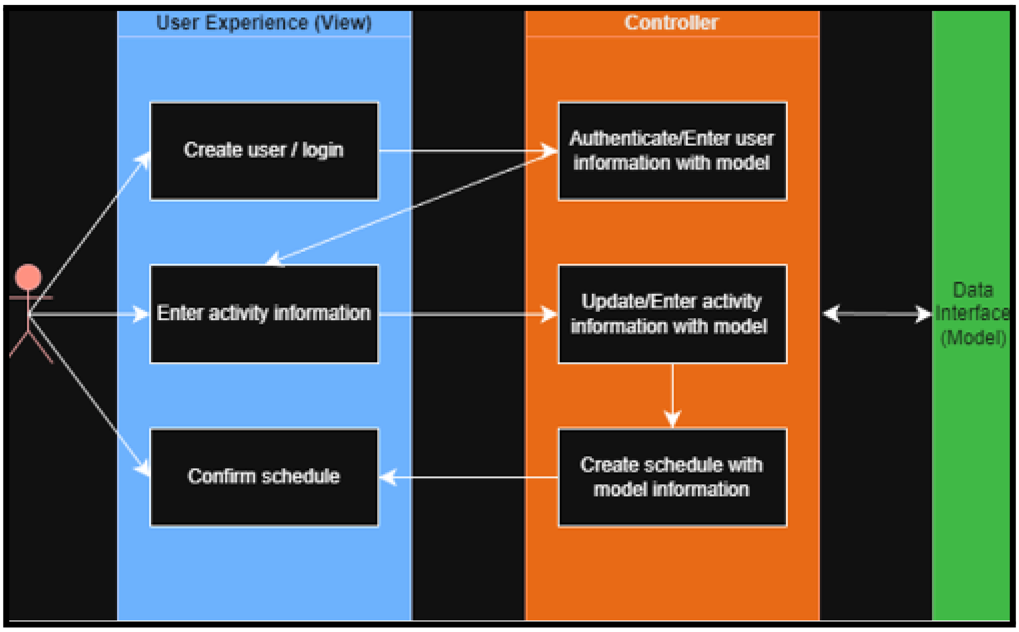


CreateUserPage	
Responsabilities	**Collaborators**
• UI to handle entering of new user information • Passes user information to database	• User

System Interaction With Environment

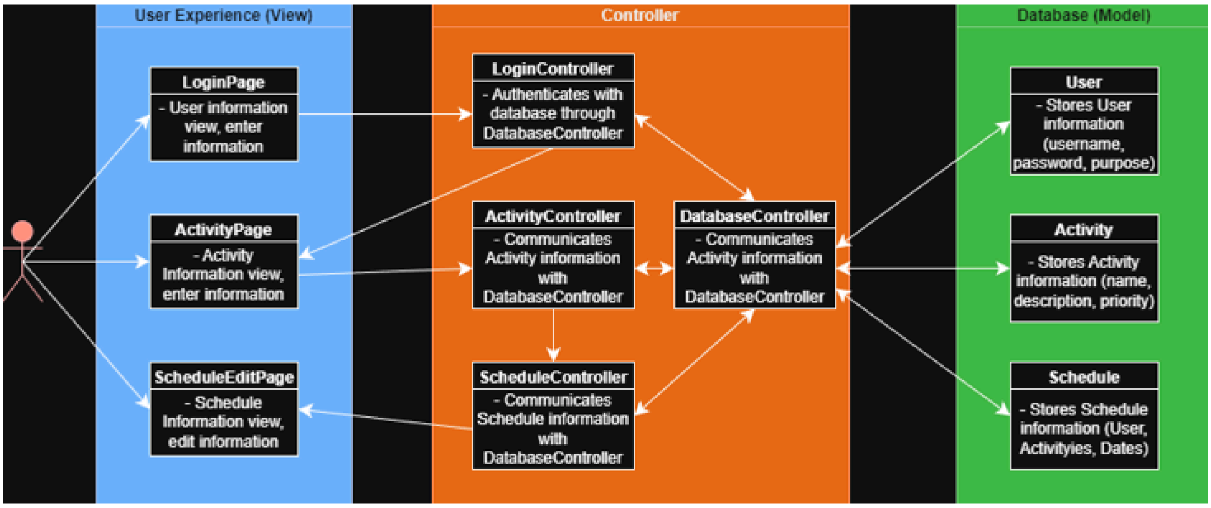
The system requires the Java 21 JDK, Java Swing (& SQL I guess? Edit Later) in order to run on all platforms (Windows, Mac, Linux). There is no internet connection required, and there is no credential/account creation required (currently) in order to use the app to its full potential.

Architecture Of System


User Experience (View)
Controller
Create user / login
Authenticate/Enter user information with model
Enter activity information
Update/Enter activity information with model
Data Interface (Model)
Confirm schedule
Create schedule with model information

On a high level, each page will transmit data entered through controllers to the database and back to the page when necessary. The database will be receiving and giving information at each stage to allow flow to the next through the controllers.

System Decomposition


User Experience (View)
Controller
Database (Model)
LoginPage
- User information view, enter information
LoginController
- Authenticates with database through DatabaseController
User
- Stores User information (username, password, purpose)
ActivityPage
- Activity Information view, enter information
ActivityController
- Communicates Activity information with DatabaseController
DatabaseController
- Communicates Activity information with DatabaseController
Activity
- Stores Activity information (name, description, priority)
ScheduleEditPage
- Schedule Information view, edit information
ScheduleController
- Communicates Schedule information with DatabaseController
Schedule
- Stores Schedule information (User, Activityies, Dates)

Each Users, Activities, and Schedules each have a Page for interaction, a Model class for its information, a Controller to handle interaction between the Page and the DatabaseController. The DatabaseController exclusively handles interaction between information sent to it, and the Model based on what was passed to it. This generic format allows either Users, Activities, or Schedules to be handled properly.