Design Document for SeaSalt

Group UG-09
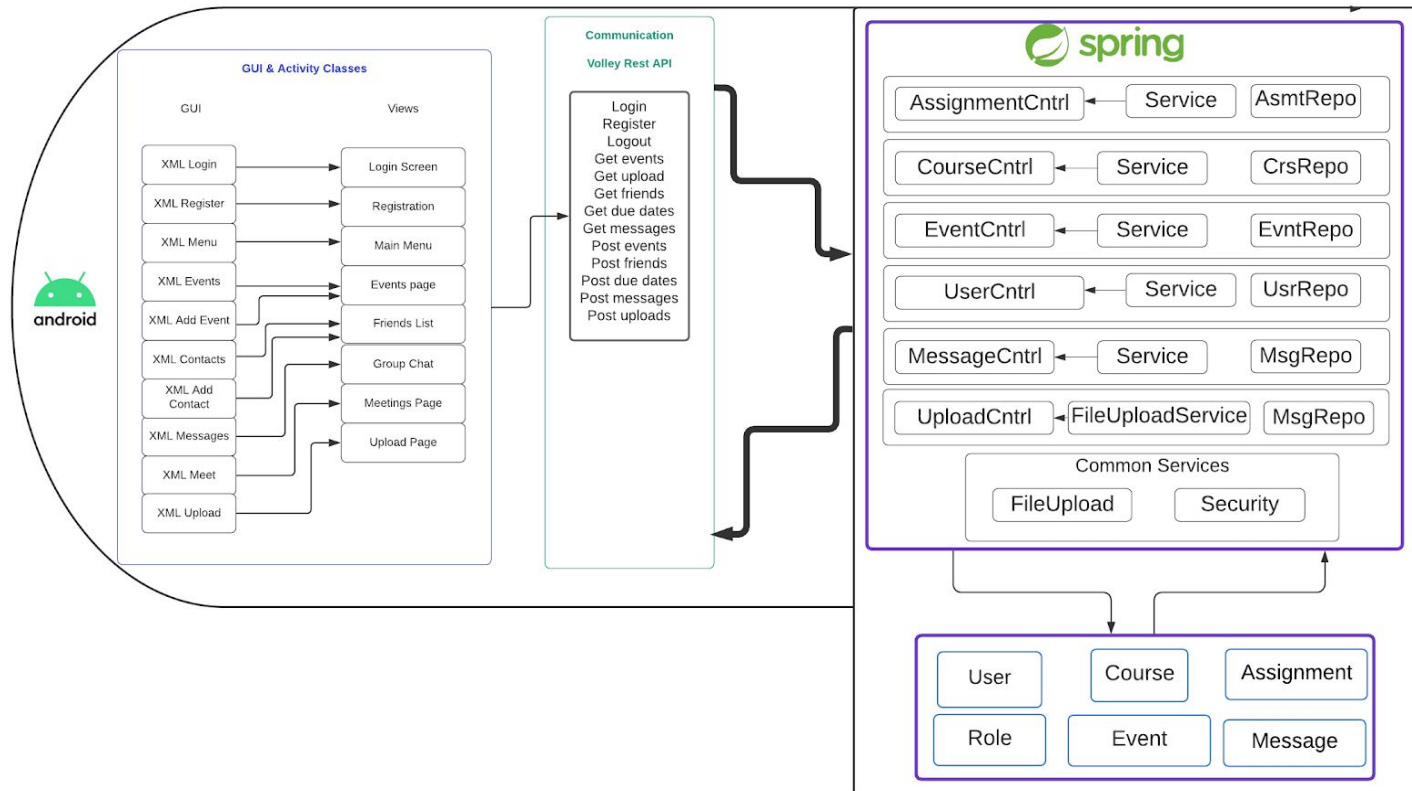
Chandler Jurenic: 100% contribution

William Blanchard: 100% contribution

Aaron Goff: 100% contribution

Edward Mahone: 100% contribution

Sea Salt By UG_09

GUI & Activity Classes

GUI

- XML Login
- XML Register
- XML Menu
- XML Events
- XML Add Event
- XML Contacts
- XML Add Contact
- XML Messages
- XML Meet
- XML Upload

Views

- Login Screen
- Registration
- Main Menu
- Events page
- Friends List
- Group Chat
- Meetings Page
- Upload Page

android

Communication
Volley Rest API

- Login
- Register
- Logout
- Get events
- Get upload
- Get friends
- Get due dates
- Get messages
- Post events
- Post friends
- Post due dates
- Post messages
- Post uploads

spring

Controller		Repository
AssignmentCntrl	← Service	AsmtRepo
CourseCntrl	← Service	CrsRepo
EventCntrl	← Service	EvntRepo
UserCntrl	← Service	UsrRepo
MessageCntrl	← Service	MsgRepo
UploadCntrl	← FileUploadService	MsgRepo

Common Services

- FileUpload
- Security

- User
- Course
- Assignment
- Role
- Event
- Message

Design Descriptions

Android GUI/Activity Classes

Seasalt mobile application will utilize the 8 activity classes with corresponding xml files plus additional xml files that are listed in the block diagram. These activity classes will be used to interact with the user and take in user commands that then implement unique versions of our communication to send the user commands data to the back end. Helpers were unable to be implemented for communication since each activity requires different data from the hashmaps.

Android Communication

The communication interface within our java project uses Volley Rest Api to take a string request from the user via front end and sends them to the server using hashmaps that are stored in the backend. This will be used to send any data that requires a user to store or get input from the server.

Spring Controllers

We have created an API for Spring that allows the frontend to interact with the information stored in the server's database. Some controllers consume multiple services, such as the upload and user controllers, so that different objects can interact with each other when extracting information such as what classes a student is in. All controllers have functions that allow the frontend to access entity information, including the details of that particular entity along with all other entities that it is associated with (ie. what events are associated with a certain assignment). The user controller allows for the addition and deletion of courses assigned to a particular user, as well as the setting of a user's role. The upload controller has functions for the uploading of user profiles, events, courses and assignments. The message controller has a function to post messages to the message board. The course controller has functions to add and edit courses in the database. The assignment controller has a function that allows the frontend to edit assignment details. The event controller has functions that allow the frontend to add and delete users and assignments to events, add events to the database, and edit event details. We have also implemented a service for server security that makes sure the user is not accessing invalid https and determines whether the user is authorized to access certain information.

Backend API:

http://cs309.rowhammer.xyz/swagger-ui.html