Milestone Application

Jordan Mower
Jonathan Hua
Jordan Nienaber
Nick Luckey

Functional Obligations From SRS

- **3.2.1.1 Functional Requirement 1 - Registration** ✓
- **3.2.1.2 Functional Requirement 2 - Log In** ✓
- **3.2.1.3. Functional Requirement 3 - Add Course** ✓
- **3.2.1.4 Functional Requirement 4 - Delete Course** ✓
- **3.2.1.5 Functional Requirement 5 - Edit Course** ✓
- **3.2.1.6 Functional Requirement 6 - Add Task** ✓
- **3.2.1.8 Functional Requirement 8 - View Task** ✓
- **3.2.1.9 Functional Requirement 9 - Delete Task** ✓
- **3.2.1.10 Functional Requirement 10 - Complete a task** ✓
- **3.2.1.11 Functional Requirement 11 - Subscribe to Course** ✓
- **3.2.1.12 Functional Requirement 12 - Search for Courses** ✓

Tools & Technologies

- AWS Amplify
    - Provided the backbone of our applications communication module.
    - GraphQL Schema Definitions using a NOSQL database.
    - User Authentication and management using Amazon Cognito UserPools.

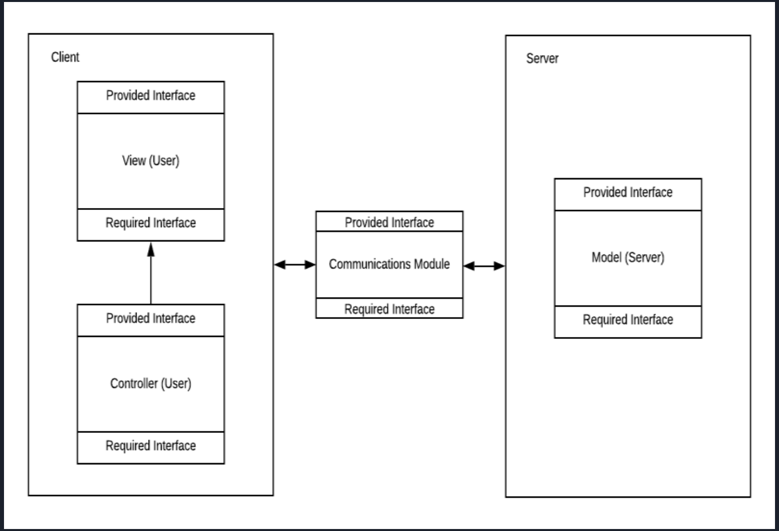
- Android Studio
    - IDE for developing android applications.
    - Provided easy to use XML Layouts for UI development.
    - Easy to add dependencies from third party developers.

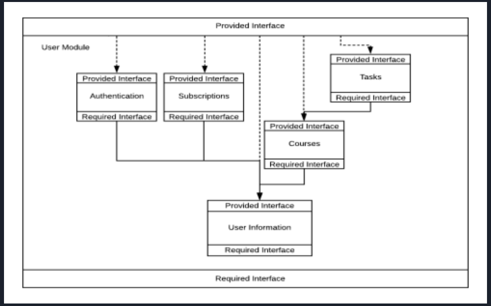
Alterations

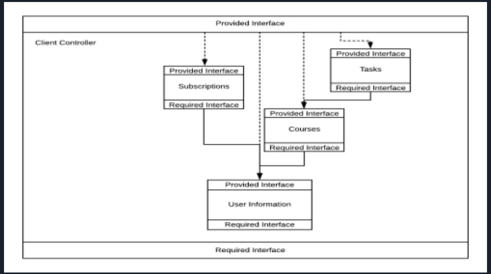
- The user now only needs a username, given name, valid email, password, and phone number. The school defaults to CSUSM, and the rest of the required information is optional and accessed from the Profile page.

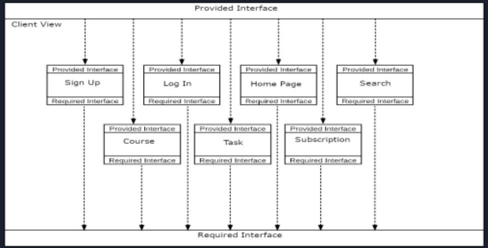
- Removed some restrictions to unnecessarily tight controls over course and task names.

Architectural Obligations









Challenges

- Team Challenges
  - No prior mobile development experience.
  - No experience with cloud based hosting.
  - No experience with AWS Amplify.
  - No experience with NoSQL & GraphQL.
- Project Challenges
  - Transfer of course/task/user information between Android activities.
    - Solution: Singleton design pattern for static storage of information related to user.
    - Drawback: Required restructuring a lot of code.

Frontend Implementation: Activities/XML & Controllers

View

- TaskViewActivity.java
- TaskViewAdapter.java
- UpcomingTaskAdapter.java
- CourseMenuActivity.java
- EditCourseActivity.java
- EditTaskActivity.java
- MainActivity.java
- ProfileActivity.java
- SubscriptionsActivity.java
- SubscriptionsAdapter.java
- AddCourseActivity.java
- AddTaskActivity.java
- AuthenticationActivity.java

Controllers

- ClientFactory.java
- CourseController.java
- UserDataController.java
- TaskController.java

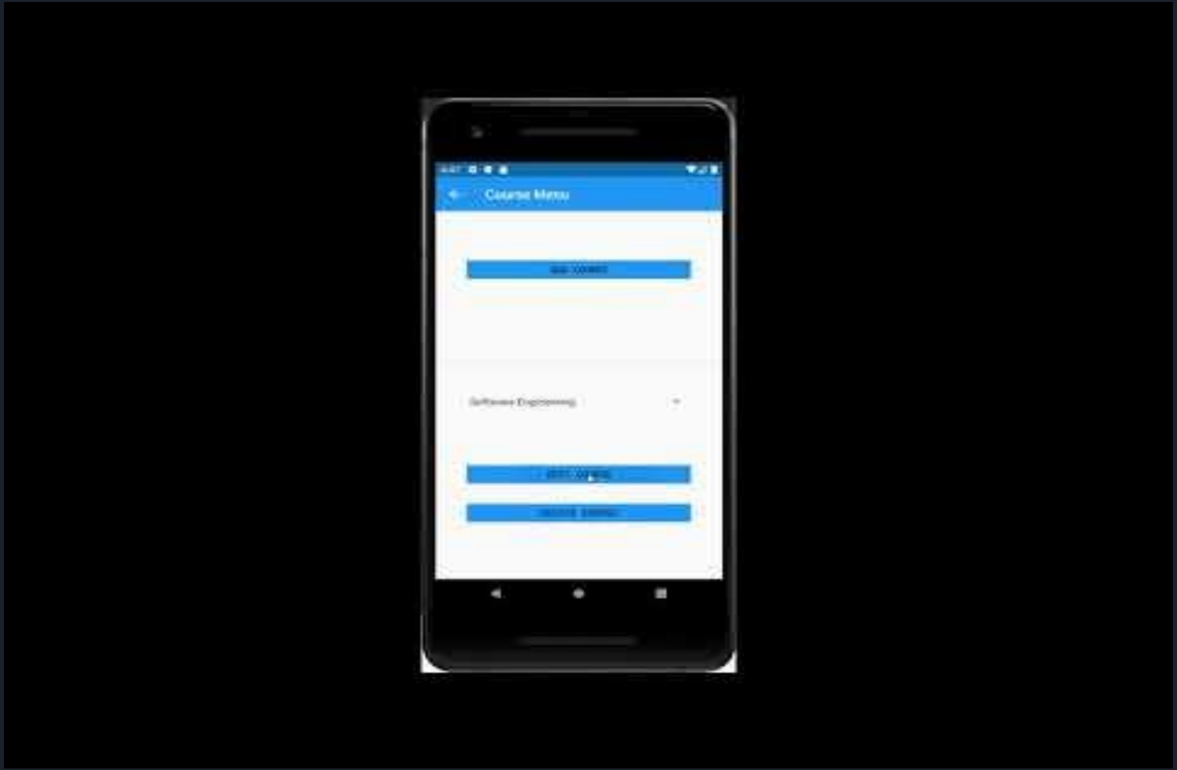
Backend Implementation:GraphQL Schema

```
type Task @model {
  id: ID!
  author: String!
  course: Course @connection(name:
"CourseTasks")
  coursename: String!
  title: String!
  duedate: String!
  percentage: Float!
  priority: String!
  comments: String!
  completed: Boolean!
}
```

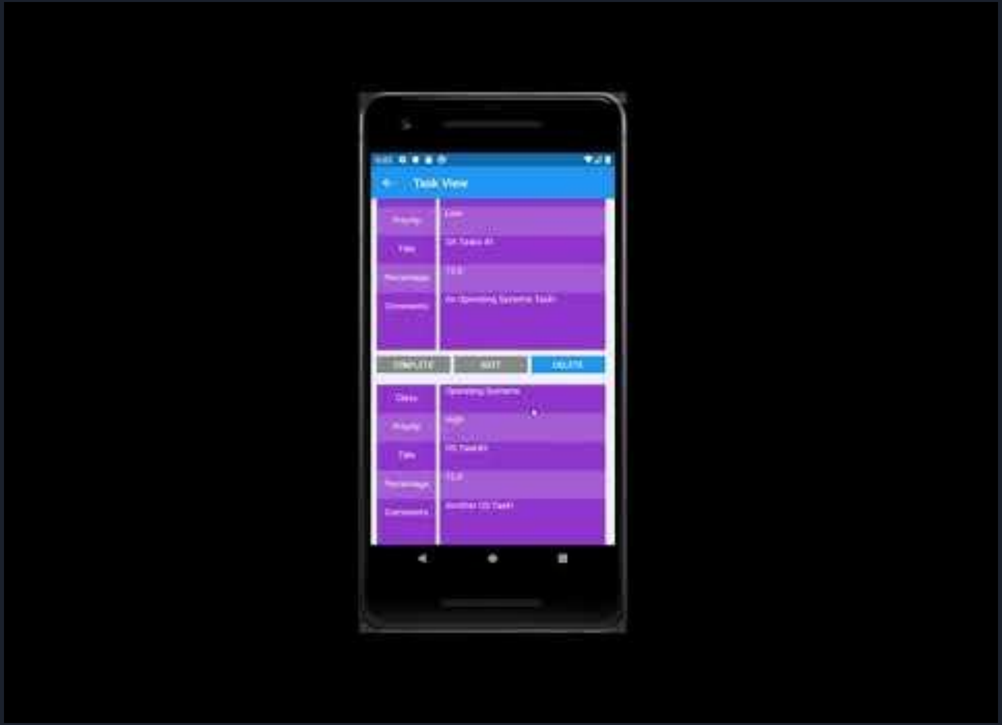
```
type Course @model {
  id: ID!
  coursename: String!
  instructor: String!
  color: String!
  meetingdays: String!
  author: String!
  tasks: [Task] @connection(name:
"CourseTasks")
}
```

```
type UserData @model {
  id: ID!
  username: String!
  schoolname: String!
  birthday: String!
  grade: String!
  subscriptions: [String]!
  firstVisit: Boolean!
}
```

A Demonstration of a new user!



An Existing User Demonstration



Task View

Priority	Low
Title	[illegible]
Percentage	[illegible]
Comments	An Operating Systems Task

Class	Operating Systems
Priority	High
Title	[illegible]
Percentage	[illegible]
Comments	Another OS Task!

From the Milestone Development Team:

Thank you!