



Ampact

Sprint 1 System Design Document

Team InsertNameHere

Luoliang Cai

David Tan

Lara Kristine Gomez

Christopher Suh

Raymond Ma

Annas Rahuma

Hui Zhi

Table of Contents

Frontend CRC Cards	3
Backend CRC Cards	8
System Architecture	10

Frontend CRC Cards

Class Name: App.js	
Parent Class (if any): Subclasses (if any):	
Responsibilities <ul style="list-style-type: none">• Manages the routing for the frontend web app	Collaborators <ul style="list-style-type: none">• React• React Router• Landing• Login• Registration• authService.js

Class Name: Landing.js	
Parent Class (if any): Subclasses (if any):	
Responsibilities <ul style="list-style-type: none">• Renders the landing page of the app	Collaborators <ul style="list-style-type: none">• React Router

Class Name: Login	
Parent Class (if any): Subclasses (if any):	
Responsibilities <ul style="list-style-type: none">• Renders the login page for the application	Collaborators <ul style="list-style-type: none">• controller.js

Class Name: Registration	
Parent Class (if any): Subclasses (if any):	
Responsibilities <ul style="list-style-type: none">• Renders the registration page for the application	Collaborators <ul style="list-style-type: none">• controller.js

Class Name: Profile (webpage)	
Parent Class (if any): Subclasses (if any):	
Responsibilities <ul style="list-style-type: none"> • Renders the profile page for the application 	Collaborators <ul style="list-style-type: none"> • controller.js

Class Name: NavigationBar	
Parent Class (if any): Subclasses (if any):	
Responsibilities <ul style="list-style-type: none"> • Renders the navigation bar for the application 	Collaborators <ul style="list-style-type: none"> • controller.js

Class Name: authService.js	
Parent Class (if any): Subclasses (if any):	
Responsibilities <ul style="list-style-type: none"> • Handles the user login, signout, and register after calling the backend 	Collaborators <ul style="list-style-type: none"> •

Class Name: controller.js	
Parent Class (if any): Subclasses (if any):	
Responsibilities <ul style="list-style-type: none"> • Sends HTTP requests to the backend and processes the response and errors 	Collaborators <ul style="list-style-type: none"> •

Class Name: Groups.js	
Parent Class (if any): Subclasses (if any):	
Responsibilities <ul style="list-style-type: none"> • Displays the group name and bio in a small area 	Collaborators <ul style="list-style-type: none"> •

Class Name: Group.js	
Parent Class (if any): Subclasses (if any):	
Responsibilities <ul style="list-style-type: none"> Adds and removes members of the group 	Collaborators <ul style="list-style-type: none"> controller.js

Class Name: GroupsList.js	
Parent Class (if any): Subclasses (if any):	
Responsibilities <ul style="list-style-type: none"> Displays other groups in a list 	Collaborators <ul style="list-style-type: none">

Class Name: GroupProfile.js	
Parent Class (if any): Subclasses (if any):	
Responsibilities <ul style="list-style-type: none"> Display's the group's profile 	Collaborators <ul style="list-style-type: none"> Group GroupMemberList React Router NavigationBar

Class Name: GroupProfileCreate.js	
Parent Class (if any): Subclasses (if any):	
Responsibilities <ul style="list-style-type: none"> Sends HTTP requests to the backend and processes the response and errors 	Collaborators <ul style="list-style-type: none"> Group.js GroupMemberList controller.js GroupsList NavigationBar

Class Name: GroupProfileEdit.js	
Parent Class (if any): Subclasses (if any):	
Responsibilities <ul style="list-style-type: none"> • Renders the page to edit a group 	Collaborators <ul style="list-style-type: none"> • NavigationBar

Class Name: GroupMember.js	
Parent Class (if any): Subclasses (if any):	
Responsibilities <ul style="list-style-type: none"> • Shows a member's name and role in a group 	Collaborators

Class Name: GroupMemberList.js	
Parent Class (if any): Subclasses (if any):	
Responsibilities <ul style="list-style-type: none"> • Shows a the list of a group's members 	Collaborators <ul style="list-style-type: none"> • GroupMember

Backend CRC Cards

Class Name: UserSchema	
Parent Class (if any): Subclasses (if any):	
Responsibilities <ul style="list-style-type: none">• Defines which collection to save group information• Defines how queries are retrieved from the database for groups• Defines how group data should be structured in database	Collaborators

Class Name: GroupSchema	
Parent Class (if any): Subclasses (if any):	
Responsibilities <ul style="list-style-type: none">• Defines which collection to save group information• Defines how queries are retrieved from the database for groups• Defines how group data should be structured in database	Collaborators

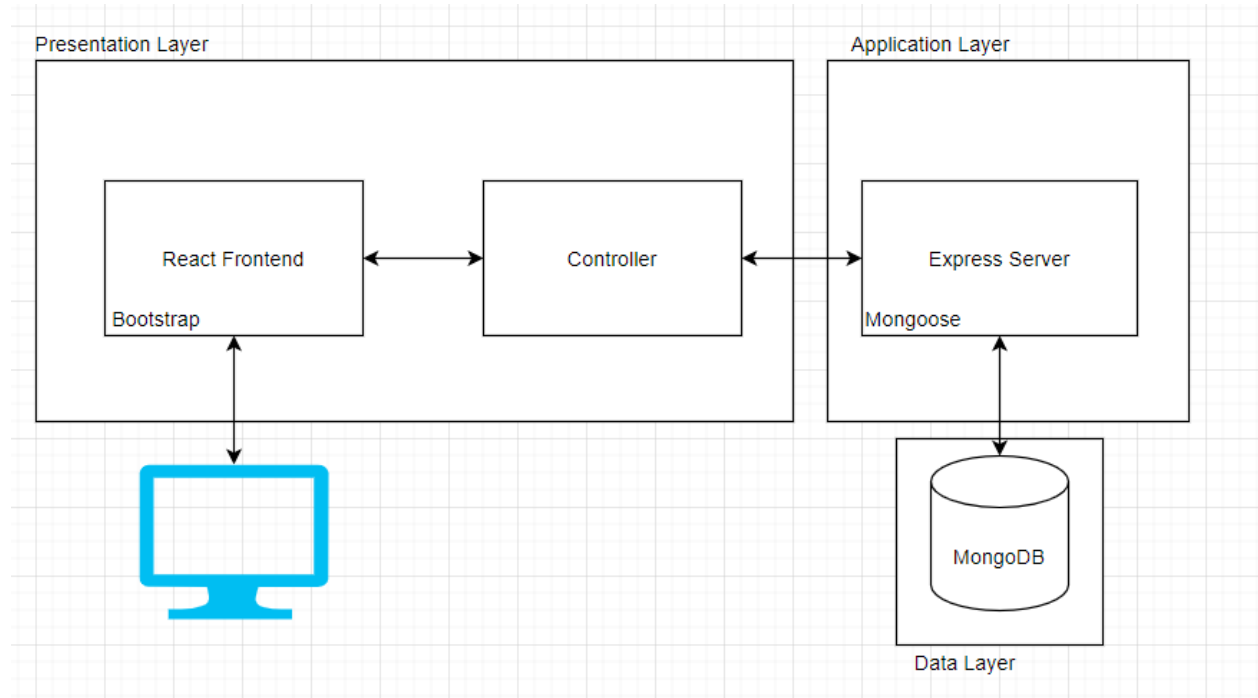
Class Name: EntrepreneurProfileSchema	
Parent Class (if any): Subclasses (if any):	
Responsibilities <ul style="list-style-type: none">• Defines which collection to save entrepreneur information• Defines what queries from entrepreneur database should contain• Defines how entrepreneur data should be saved in database	Collaborators

Class Name: InstructorProfileSchema	
Parent Class (if any): Subclasses (if any):	
Responsibilities <ul style="list-style-type: none"> • Defines which collection to save instructor information • Defines what queries from instructor database should contain • Defines how instructor data should be saved in database 	Collaborators

Class Name: PartnerProfileSchema	
Parent Class (if any): Subclasses (if any):	
Responsibilities <ul style="list-style-type: none"> • Defines which collection to save Partner profile information • Defines what queries from partner database should contain • Defines how partner data should be saved in database 	Collaborators

System Architecture

This application will be using the 3-tiered architecture.



React Frontend: The presentation layer. This will contain the user interface for the application

Controller: Contains all the API requests to retrieve data from the application layer for the frontend to call.

Express Server: Handles the logic of the API requests and returns a response or error to the frontend. Connected to MongoDB using Mongoose.

MongoDB: The data layer. Where all information is stored (User, Profiles, Groups, Images)