

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON**

**ARCHITECTURAL DESIGN SPECIFICATION
CSE 4316: SENIOR DESIGN I
FALL 2022**



**TEAM CORPS CATCH
CORPS CATCH**

KEVIN LE
KOUASSI BROU
OLUWASEETOFUNMI KOMOLAFE
BIJAN SAUD
JIANLIANG LIU

REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	10.26.2022	KL	document creation
0.2	11.02.2022	KL, KB, OK, BS, JL	complete draft
0.3	11.07.2022	KL	official release

CONTENTS

1	Introduction	7
2	System Overview	8
2.1	Front End/ UI Layer Description	8
2.2	Back end Layer Description	8
2.3	Database Layer Description	9
3	Subsystem Definitions & Data Flow	10
4	Front End Layer Subsystems	11
4.1	Sign-up	11
4.2	Login	12
4.3	Map Page	13
4.4	Question Page	14
4.5	Collection	15
5	Back End Layer Subsystems	17
5.1	Authentication	17
5.2	Map API	18
5.3	Get Question API	18
5.4	Get Answer	20
5.5	Get Coins	21
6	Database Layer Subsystems	22
6.1	User Info	22
6.2	Trail Q/A	22

LIST OF FIGURES

1	A simple architectural layer diagram	8
2	A simple data flow diagram	10
3	Front-end subsystem description diagram	11
4	Front-end subsystem description diagram	12
5	Front-end subsystem description diagram	13
6	Home subsystem diagram	14
7	Front-end subsystem description diagram	15
8	Back-end subsystem description diagram	17
9	Example subsystem description diagram	19
10	Example subsystem description diagram	19
11	Example subsystem description diagram	20
12	Example subsystem description diagram	21
13	Database layer subsystem diagram	22
14	Example subsystem description diagram	23

LIST OF TABLES

2	Sign-Up Subsystem Interfaces	12
3	Login Subsystem Interfaces	13
4	Map Subsystem Interfaces	14
5	Question Subsystem Interfaces	15
6	Collection Subsystem Interfaces	16
7	Authentication Subsystem Interfaces	18
8	Map Subsystem Interfaces	18
9	Back End and Database Connection Subsystem Interfaces	20
10	Back End and Database Connection Subsystem Interfaces	21
11	Back End and Database Connection Subsystem Interfaces	21
12	User Info Subsystem Interfaces	22
13	Trail Q Subsystem Interfaces	23

1 INTRODUCTION

The Corps Catch application would make exploring the outdoor trails in a Corps facility more fun and engaging. Its goal is to provide entertainment and education to its users, and this is achieved by asking increasingly difficult questions based on the current subject of choice during the hiking trail. When all questions are answered correctly the player is awarded a digital Corps Coin which they can view in their collection. This reward system is based on the actual coins that outstanding personnel receive in the Corps. The aim is to collect as many coins as possible along the hiking trail by answering questions correctly.

Key requirements for this project include tracking the user's location and have questions pop up when they are within a certain radius of the question, the award of a corp coin if they get all the questions right on the trail, and keeping user information including login and any corp coins they have stored in a database.

The scope of the project is documented by the Project Charter, System Requirements Specification, and Architectural Design Specification. These will include the functionalities and how the app will be designed.

2 SYSTEM OVERVIEW

The Corps Catch Application consists of three primary layers: The Front End/User Interface layer, the Back End/Data Processing layer, and the Database Layer. These layers, their functions, and their interactions with each other are described as follows:

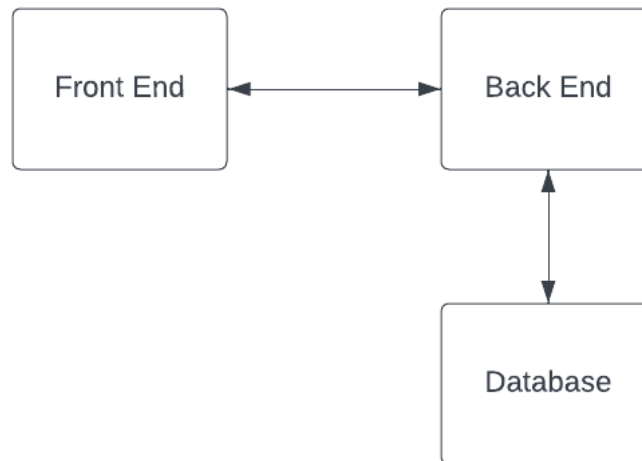


Figure 1: A simple architectural layer diagram

2.1 FRONT END/ UI LAYER DESCRIPTION

This Layer consists of the visual aspects of the applications and is essentially everything that the users see and interact with. This layer also handles the inputs and placements of the outputs. In The Corps Catch Application, this layer should allow users to interact with and navigate to all the pages of the app, be visually appealing, and also easy to utilize. An example of the flow of the UI elements would be traversing from the Login Page to the Home Page and then the Coin Collection Tab. The front end will interact with the back end through any action requiring a change in state or publishing of information. For instance, clicking a button to go to another page or submitting the form the user gets when they are logging in.

2.2 BACK END LAYER DESCRIPTION

This layer will pass information between the front end and the database. Since information from the front end cannot be directed to the database, this layer will be used to communicate between the two. In the sign-up and login function of the application, this layer will get the information the user has entered and once the user presses submit, this layer gathers the information from the front end and stores it if the user is signing up, and validates whether the user exists in the database if the user is logging in. When the user is answering a question this layer gets the answer choice the user makes, and validates whether the answer matches the answer in the database. This layer will essentially be used to validate or execute necessary implementations within the application. Subsystems for this layer include front-end connection and database connection.

2.3 DATABASE LAYER DESCRIPTION

This layer will store all user information including login information and profile information. This layer will also store questions for each trial, each question will have information including, the question, answer, difficulty, and which trail the question is for. When a user is signing up the back end communicates with this layer to make sure the user does not exist, then adds the user to the database. When a user logs in, the back end and this layer communicate to validate that the user exists, and navigate the user to the home screen if validated. When the user answers a question, a request is sent from the back end to this layer to check if the user is correct or not. If the users want to make changes to their profile, a request is sent from the back end and the selected changes will be made. Subsystems for this layer include database insertion and database query.

3 SUBSYSTEM DEFINITIONS & DATA FLOW

Our system comprises of a front-end layer and a data layer, which includes a back-end layer and a database layer.

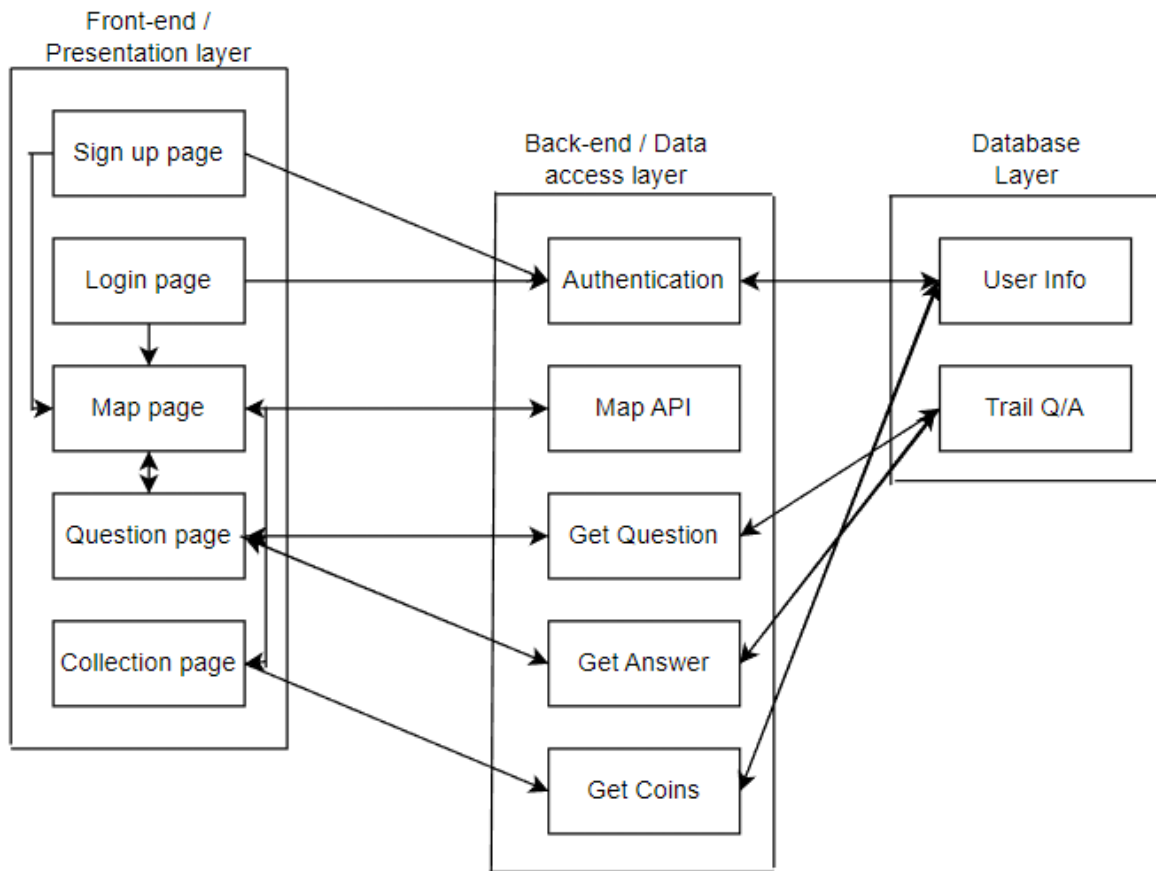


Figure 2: A simple data flow diagram

4 FRONT END LAYER SUBSYSTEMS

The front-end layer as aforementioned regards the visual aspects of the app that the user interacts with. The subsystems involved in the Front-end are Sign-up, Login, Home, and Collection.

4.1 SIGN-UP

This subsystem is important and necessary for every new user. The data they input here would be need to be accurate and remembered as this would tie the user to the progress they make such as customization, experience points, and collected corp coins. This subsystem communicates with the back end and database layers.

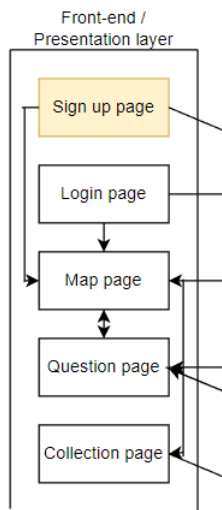


Figure 3: Front-end subsystem description diagram

4.1.1 ASSUMPTIONS

For a user to sign-up they should be a new user or want to make a new account.
They are using a working unique email address to sign up.
Password has at least one number in it.

4.1.2 RESPONSIBILITIES

This subsystem is responsible for the initial placement of users into the database and registering their account.

4.1.3 SUBSYSTEM INTERFACES

Table 2: Sign-Up Subsystem Interfaces

ID	Description	Inputs	Outputs
#1	Registering of User	First Name Last Name Username Email Phone number Password	Addition of user into the database Creation of account
#2	Registering of User	First Name Last Name Username Email Phone number Password	User enters invalid email, user is prompted to add valid email
#3	Registering of User	First Name Last Name Username Email Phone number Password	User enters email that is already in use, prompted to login

4.2 LOGIN

This subsystem connects existing users to their account and its records after authentication (username and password) is granted.

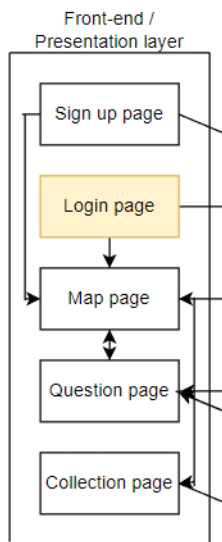


Figure 4: Front-end subsystem description diagram

4.2.1 ASSUMPTIONS

User must not be logged in.
User must have an existing account.
Accurate information is given.

4.2.2 RESPONSIBILITIES

This subsystem is responsible for maintaining user progress and connecting existing users to their records.

4.2.3 SUBSYSTEM INTERFACES

Table 3: Login Subsystem Interfaces

ID	Description	Inputs	Outputs
#1	Login	Username Password	Correct credentials, user is granted access to their account.
#1	Login	Username Password	Incorrect credentials, user is not granted access to their account.

4.3 MAP PAGE

This subsystem is the integral part of the application. It is the core to what the application does that it fetch trails and show current location in the map. All the interaction of user while using the app will be in this page.

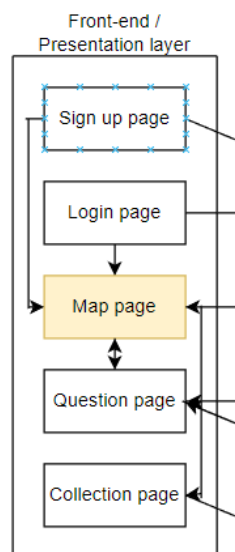


Figure 5: Front-end subsystem description diagram

4.3.1 ASSUMPTIONS

The user must be logged in.

The user should navigate to map page.

4.3.2 RESPONSIBILITIES

This subsystem when navigated shows the available trials. It also navigates the user while in the trail updating automatically.

4.3.3 SUBSYSTEM INTERFACES

Table 4: Map Subsystem Interfaces

ID	Description	Inputs	Outputs
#1	Map	User clicks the Maps tab	Available trails are shown
#1	Map	User starts trail	Map updates as user moves through the trail

4.4 QUESTION PAGE

This subsystem displays the users question when triggered while on trail.

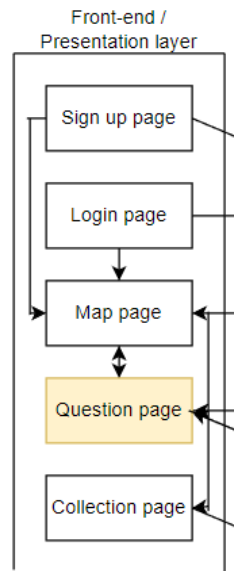


Figure 6: Home subsystem diagram

4.4.1 ASSUMPTIONS

User must be logged in.

4.4.2 RESPONSIBILITIES

This subsystem is responsible for showing questions while on the trail. The screen is popped by a question when the user reaches the destination.

4.4.3 SUBSYSTEM INTERFACE INTERFACES

Table 5: Question Subsystem Interfaces

ID	Description	Inputs	Outputs
#1	Question	User is prompted to answer a question	Experience points are given based on accuracy of question.

4.5 COLLECTION

This subsystem shows all the corps coins that a user has collected and also shows the current experience points that the user has.

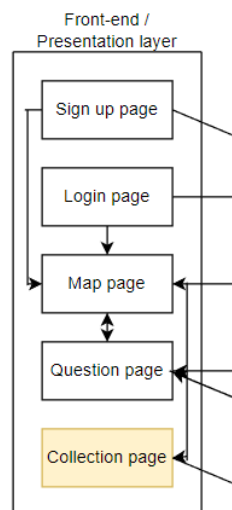


Figure 7: Front-end subsystem description diagram

4.5.1 ASSUMPTIONS

User must be logged in.

The user must toggle the collection page button.

4.5.2 RESPONSIBILITIES

This subsystem is responsible for displaying user records such as number of coins collected as well as experience points

4.5.3 COLLECTION SUBSYSTEM INTERFACE

This subsystem also interacts with the database and back-end to retrieve relevant information.

Table 6: Collection Subsystem Interfaces

ID	Description	Inputs	Outputs
#1	Collection	User clicks Collections Page	All collected corps coins are displayed User's experience points displayed

5 BACK END LAYER SUBSYSTEMS

The back end comprises of a connection to the front end, a connection to the database.

5.1 AUTHENTICATION

Back-end grabs data that user has entered and will check with the database whether or not the user exists.

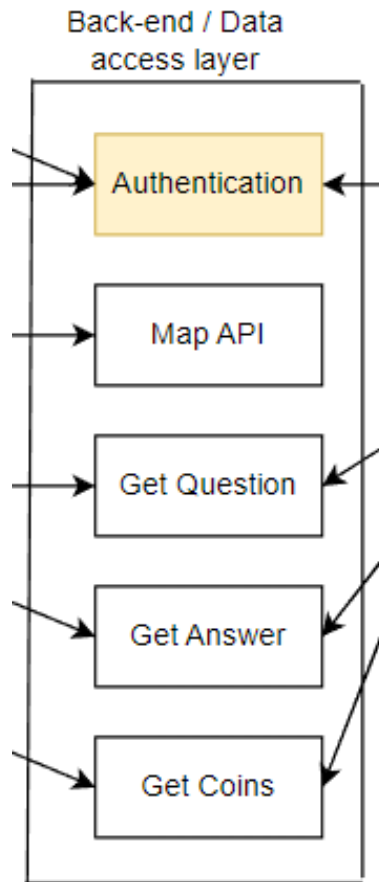


Figure 8: Back-end subsystem description diagram

5.1.1 ASSUMPTIONS

Proper queries will be made to ensure user does / does not exist.

5.1.2 RESPONSIBILITIES

User should be able to log in with correct credentials and should be denied login with invalid user and password.

5.1.3 SUBSYSTEM INTERFACES

Table 7: Authentication Subsystem Interfaces

ID	Description	Inputs	Outputs
#1	User signs up from signup page	username password email name age	email and user-name not in user sign up success
#2	User signs up from signup page (with existing email)	username password email name age	email in user sign up failed
#3	User logs in from login page	username password	Username and password match login success
#4	User logs in from login page (invalid)	username password	Username and password do not match login fail

5.2 MAP API

The back-end layer contains Map API which will provide access to real time Maps, and location will be shown on map.

5.2.1 ASSUMPTIONS

There will be sufficient code that enables user to see their location map and question should pop up as intended.

5.2.2 RESPONSIBILITIES

Show map to user, including user location.

5.2.3 SUBSYSTEM INTERFACES

Table 8: Map Subsystem Interfaces

ID	Description	Inputs	Outputs
#1	User location	User latitude and longitude	User on map

5.3 GET QUESTION API

Back-end gets question for user based on user's age, location on trail and which trail the user is on.

5.3.1 ASSUMPTIONS

There will be sufficient code to get corresponding question based on user location.

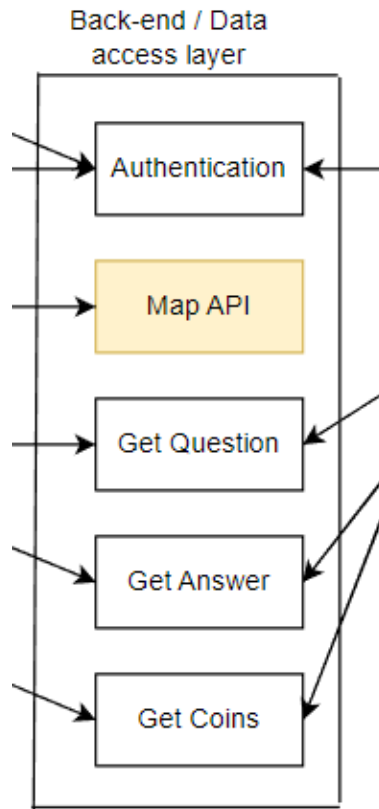


Figure 9: Example subsystem description diagram

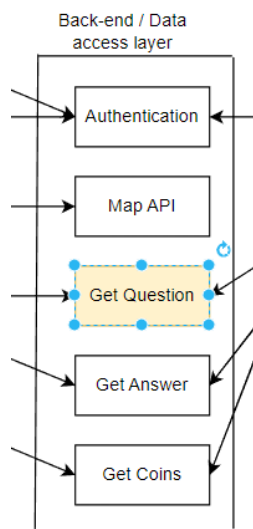


Figure 10: Example subsystem description diagram

5.3.2 RESPONSIBILITIES

Get correct question for user, so the question should correspond to the location and age of the user.

5.3.3 SUBSYSTEM INTERFACES

Table 9: Back End and Database Connection Subsystem Interfaces

ID	Description	Inputs	Outputs
#1	User clicks question on screen	User age (< 13), location, trail id	Question (easy)
#2	User clicks question on screen	User age (13-17), location, trail id	Question (med)
#3	User clicks question on screen	User age (>= 18), location, trail id	Question (easy)

5.4 GET ANSWER

The back-end gets the answer to the question the user is on.

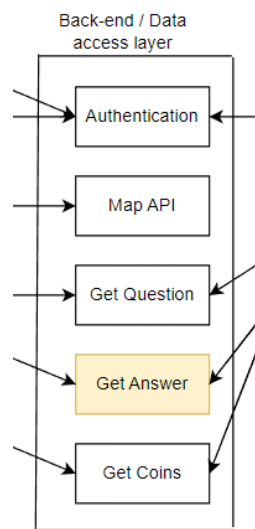


Figure 11: Example subsystem description diagram

5.4.1 ASSUMPTIONS

There will be queries available to get correct answer to question.

5.4.2 RESPONSIBILITIES

Back-end gets correct answer and compares it to the answer the user chose, if matches correct else incorrect.

5.4.3 SUBSYSTEM INTERFACES

Table 10: Back End and Database Connection Subsystem Interfaces

ID	Description	Inputs	Outputs
#1	Get answer to question	Trail question id	Answer to question

5.5 GET COINS

This part of the back-end works with the database to retrieve the user's corp coins earned to display them to user.

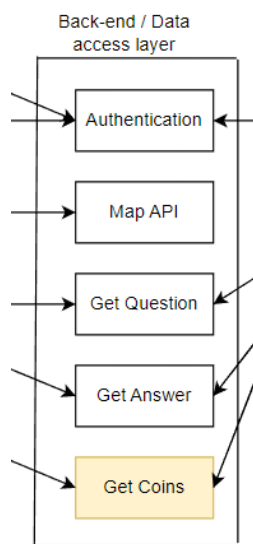


Figure 12: Example subsystem description diagram

5.5.1 ASSUMPTIONS

Proper queries will be implemented to get user's coins.

5.5.2 RESPONSIBILITIES

Retrieve correct user and get corp coins obtained from trails.

5.5.3 SUBSYSTEM INTERFACES

Table 11: Back End and Database Connection Subsystem Interfaces

ID	Description	Inputs	Outputs
#1	Get Coins	User id	Corp Coins obtained

6 DATABASE LAYER SUBSYSTEMS

This layer is where all information and data including user information/ login and trail questions are stored. The database layer is split between two subsystems user info and trail q/a.

6.1 USER INFO

This part of the database holds user information including username, email, password, and corp coins earned.

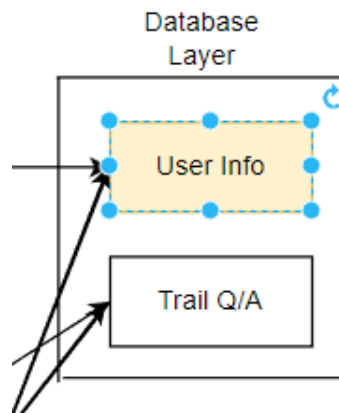


Figure 13: Database layer subsystem diagram

6.1.1 ASSUMPTIONS

Queries will not contain any errors when called. Queries will not cause any errors.

6.1.2 RESPONSIBILITIES

This part of the database will store each user's information, and the responsibilities of this part of the database include storing the right data in the right parts of the database. The information should also be identical to what the user inputs.

6.1.3 SUBSYSTEM INTERFACES

Table 12: User Info Subsystem Interfaces

ID	Description	Inputs	Outputs
#1	Authentication	Username and Password	User info
#2	Authentication	Invalid Username and Password	Error message
#2	Coin Retrieval	User info	Coins collected

6.2 TRAIL Q/A

This part of the database holds trail questions, answers, difficulty level, and which trail the question is on.

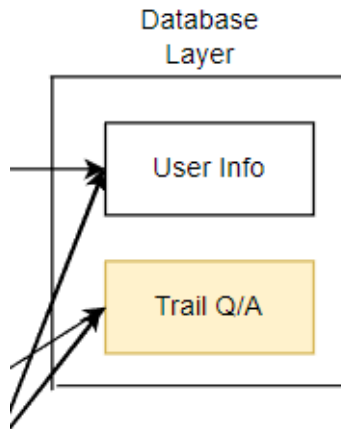


Figure 14: Example subsystem description diagram

6.2.1 ASSUMPTIONS

Trail questions and answers will have a place in database and queries to insert and retrieve data work as intended.

6.2.2 RESPONSIBILITIES

The responsibility of the database insertion is making sure that the trail questions and answers are stored properly with each question having the correct answer. Question difficulty and corresponding trail must be accurate also.

6.2.3 SUBSYSTEM INTERFACES

Table 13: Trail Q Subsystem Interfaces

ID	Description	Inputs	Outputs
#1	Question retrieval	User and Trail info	Question info
#2	Answer to Question	Question request	Answer info

REFERENCES