

CS361: Assignment 6: Sprint 2 Plan (for Microservice A)

# Overview

Plan the microservice you’ll make for your teammate(s). That includes defining how to **request** and **receive data** from the microservice.

# Instructions

Complete each item below by replacing the highlighted text (**Usability note:** double-click the text to select it).

1. What is your **Sprint Goal**? (e.g., fully implement the spell-checker microservice) The Sprint Goal must clearly communicate what the microservice will do.

|  |
| --- |
| *Fully implement Event discovery app microservice* |

1. Define **at least three user stories** for this Sprint. Provide your user stories and their functional and non-functional acceptance criteria (and associated quality attributes).

**Requirements for Microservice A:**

* You must implement at least three user stories.
* Each user story must have a name.
* Each user story must use the “As a… I want to... so that…” format.
* Each user story must have at least one functional acceptance criterion.
* All functional acceptance criteria must use the “Given… when… then…” format.
* At least one of the user stories must have an associated quality attribute and non-functional acceptance criterion.

**First user story**

|  |
| --- |
| (Front of index card)  *Group Chat*  As a user I want the ability to communicate with other individuals in my neighborhood that enjoys the same hobbies as me. |
| (Back of index card)  **Acceptance criteria**  Functional requirements   * *Given the normal average people sends and receives a text message is 41.5 texts messages per day, then the app must maintain a storage capacity with enough database to sustain such this app feature when user sends a message*   Quality attributes & Non-functional requirements   * *Responsiveness: It takes an average of 6 seconds for a message to be sent over the network. To maintain effective responsiveness the message for the chat feature should be sent within an interval of 5-10 seconds.* |

**Second user story**

|  |
| --- |
| (Front of index card)  *Tech Support*  As an IT admin, I want the developers to provide documentation of the app normal behavior and supported features for excellent customer support. |
| (Back of index card)  **Acceptance criteria**  Functional requirements   * *Given the app is well written using pseudocodes and user research analysis, when the IT admin requests for assistance in troubleshooting an issue then the information will be provided quicker and in a more descriptive manner.*   Quality attributes & Non-functional requirements   * *Reliability: Developers should provide easy and accessible documentation to IT administrators for the supported OS systems (iOS and Android) in order to quickly resolve customer facing inquiries (e.g. UI, APIs, User Account troubleshooting issues, etc.) and establish a good user experience within 5 minutes.* |

**Third user story**

|  |
| --- |
| (Front of index card)  *Profile Page*  As a new user, I want to have the ability to edit and customize my user profile and any images that will appear to the public. |
| (Back of index card)  **Acceptance criteria**  Functional requirements   * *Given the new user will have the ability to add a profile picture and up to 6 images embedded within their profile, the user will need to allow the app to access their photo gallery, then the developer’s API can be implemented the access their local storage when they create their profile and change their local settings.*   Quality attributes & Non-functional requirements   * *Compatibility: The program code must contain compatible solution and system when the user is using the app, then the user experience will be smooth no matter which device they are using from.* |

## Take a screenshot that shows you’ve moved these user stories into a Sprint Backlog in your task management system.

|  |
| --- |
|  |

1. What kind of **communication pipe** will the microservice use? (e.g., text files, REST API)

|  |
| --- |
| *Rest API* |

1. How will other programs be able to **request data** from the microservice and what request parameters will be required? Give an example call using pseudocode or actual code.

|  |
| --- |
| Request parameters (included parameters in the request call)   **GET Request** with Query Parameters   **POST Request** with JSON Body Parameters   **Handling Request Parameters in URL** |
| Detailed description (how data will be requested)  GET Request: Parameters are usually in the URL as query parameters (e.g., ?userId=1234&page=2).  POST Request: Parameters are sent in the body as JSON data.  Path Parameters: Sometimes parameters are passed directly in the URL path (e.g., /users/1234).  Headers: Some APIs require additional headers for authentication, like Authorization or x-api-key |
| Example call (pseudocode or actual code)  import Foundation  func fetchUserById(userId: String) {  // Construct the URL with the path parameter  let urlString = "https://api.example.com/users/\(userId)"  guard let url = URL(string: urlString) else {  print("Invalid URL")  return  }    var request = URLRequest(url: url)  request.httpMethod = "GET" // Default is GET, so this is optional  // Perform the GET request  let task = URLSession.shared.dataTask(with: request) { data, response, error in  if let error = error {  print("Error: \(error.localizedDescription)")  return  }    // Handle the response data  if let data = data {  do {  if let json = try JSONSerialization.jsonObject(with: data, options: []) as? [String: Any] {  print("Response: \(json)")  } else {  print("Invalid response data")  }  } catch {  print("Error parsing JSON: \(error.localizedDescription)")  }  }  }  task.resume()  }  // Example usage  fetchUserById(userId: "1234") |

1. How will other programs be able to **receive data** from the microservice and what data will the microservice provide? Give an example call using pseudocode or actual code.

|  |
| --- |
| API Endpoint: /users/{userId} (GET)  Data Provided by the Microservice: User information in JSON format. |
| Detailed description (how data will be provided)  **User Information**: The microservice will provide details about a user, such as their name, email, age, etc.  {  "userId": "1234",  "name": "John Doe",  "email": "johndoe@example.com",  "age": 30,  "address": "1234 Elm Street, Springfield, IL"  } |
| Example call (pseudocode or actual code)  import Foundation  func fetchUserData(userId: String) {  // Construct the URL with the userId  let urlString = "http://localhost:3000/users/\(userId)"  guard let url = URL(string: urlString) else {  print("Invalid URL")  return  }  // Create a GET request  var request = URLRequest(url: url)  request.httpMethod = "GET" // GET method for fetching data  // Perform the request  let task = URLSession.shared.dataTask(with: request) { data, response, error in  if let error = error {  print("Error: \(error.localizedDescription)")  return  }  // Handle the response data  if let data = data {  do {  // Parse the JSON response  if let json = try JSONSerialization.jsonObject(with: data, options: []) as? [String: Any] {  // Extract the user data from JSON  if let userId = json["userId"] as? String,  let name = json["name"] as? String,  let email = json["email"] as? String,  let age = json["age"] as? Int,  let address = json["address"] as? String {  print("User Details:")  print("UserId: \(userId)")  print("Name: \(name)")  print("Email: \(email)")  print("Age: \(age)")  print("Address: \(address)")  }  }  } catch {  print("Error parsing JSON: \(error.localizedDescription)")  }  }  }  task.resume()  }  // Example usage  fetchUserData(userId: "1234") |

**This would be a good time to make a new repository to house the microservice.**

# Submission

PDF or Word format via Canvas.

**You must follow instructions at Modules > “Attach a Document to "Text Entry" Field”.**

# Grading

You are responsible for satisfying all criteria listed in the Canvas rubric for this assignment.

# Questions?

Please ask via Ed so that others can benefit from the answer.