New project research after the 3rd project presentation [26 May 2023]

To do list:

1. Redraw a project schedule.
2. Use agile methodology to carry out the project.
3. Update final report document daily.
4. Test the system daily as per updates made and involve the users for testing.
5. Find a new programming language to use in developing the system.
6. Review the project guidelines daily while developing the system.
7. Revisit the scope, aims, objectives and goals of the system.
8. Aims:

* Navigation
* Location-based services
* information about a location such as rooms in a building

1. Objectives:

* To develop a Geographic Information System (GIS) that can assist with navigation within the NUL campus.
* To build a platform that can provide the user with directions to desired destinations on-campus.
* Providing accurate and up-to-date maps.
* Helping users navigate.
* Enhancing user experience.
* Supporting multiple platforms.

1. Scope

* The project will include the design, implementation, testing and delivery of a software system.
* Platform should be capable of offering maps with interactive directions for the users.
* User’s pre-set location.

1. Use case diagram

* Display map (in progress, give users to test )
* Search destination (To do today)
* Display routes to destination
* Give directions to destination
* Can zoom in and out of map (in progress, give users to test)
* Can find users current location on map (To do today)
* System gives shortest route to destination

1. Keep supervisor updated as often as possible.
2. Apply as much effort in developing the project as possible and enjoy as well as loving the project.
3. Create a github account and upload project code after every update.

To create a GitHub account and upload your project code, follow these steps:

1. Go to the GitHub website ([https://github.com](https://github.com/)).
2. Click on the "Sign up" button located at the top-right corner of the page.
3. Fill in the required information, including your username, email address, and password. Choose a strong password and ensure that your username is unique and available.
4. Select a plan for your GitHub account. You can choose the free plan, which provides basic features.
5. Complete any additional steps or verifications required during the sign-up process.
6. Once you've created your GitHub account and logged in, you can create a new repository to upload your project code.
7. Click on the "+" sign in the top-right corner and select "New repository" from the drop-down menu.
8. Enter a name for your repository. This name should be descriptive and related to your project.
9. Optionally, you can provide a description for your repository to give more information about your project.
10. Choose whether you want your repository to be public (visible to everyone) or private (visible only to you or collaborators).
11. Check the box that says "Initialize this repository with a README" if you want to create an initial README file for your project. It's a good practice to include a README file with information about your project.
12. Click on the "Create repository" button to create your repository.
13. On the repository page, you will see options to clone the repository or upload files. You can choose the method that suits you best.
14. If you want to upload your project code, click on the "Upload files" button.
15. Drag and drop the project files or click on the "Choose your files" button to select the files from your computer.
16. Once the files are selected, add an optional description or message about the changes you made.
17. Click on the "Commit changes" button to upload your project files to the repository.
18. Your project code is now uploaded to your GitHub repository.

Note: It's a good practice to create a **.gitignore** file in your project to specify which files and directories should be ignored when committing changes. This helps to avoid uploading unnecessary files, such as temporary files or sensitive information.

Additionally, you may want to explore more advanced features of GitHub, such as creating branches, making pull requests, and collaborating with other developers. The GitHub documentation provides detailed information and guides on these topics.

7. Revisit the scope, aims, objectives and goals of the system. 1. Aims: ϖ Navigation ϖ Location-based services ϖ information about a location such as rooms in a building 2. Objectives: ϖ To develop a Geographic Information System (GIS) that can assist with navigation within the NUL campus. ϖ To build a platform that can provide the user with directions to desired destinations on-campus. ϖ Providing accurate and up-to-date maps. ϖ Helping users navigate. ϖ Enhancing user experience. ϖ Supporting multiple platforms. 3. Scope ϖ The project will include the design, implementation, testing and delivery of a software system. ϖ Platform should be capable of offering maps with interactive directions for the users. ϖ User’s pre-set location. 4. Use case diagram ϖ Display map ϖ Search destination ϖ Display routes to destination ϖ Give directions to destination ϖ Can zoom in and out of map ϖ Can find users current location on map ϖ System gives shortest route to destination From the above information develop a simple and user satisfying web-app navigation system in python using mapbox api and necessary libraries or framework

ChatGPT

To develop a simple and user-satisfying web-app navigation system in Python using the Mapbox API and necessary libraries or frameworks, you can follow these steps:

1. Set up the project:
   * Install the required libraries and frameworks such as Flask, Mapbox SDK, and any other dependencies you may need.
   * Create a new Python project directory.
   * Initialize a virtual environment to manage your project dependencies.
2. Create a Flask application:
   * Set up a basic Flask application with routes and views.
   * Create a route to handle the homepage or landing page of your navigation system.
3. Integrate Mapbox API:
   * Obtain an access token from the Mapbox website (<https://www.mapbox.com/>) to authenticate your API requests.
   * Use the Mapbox SDK or REST API to interact with the map services.
   * Implement functions or methods to fetch map data, directions, and location-based services from the Mapbox API using the provided access token.
4. Design and implement the user interface:
   * Create HTML templates for different pages/views, such as the homepage, search page, and route display page.
   * Use CSS and JavaScript to enhance the user interface and interactivity.
   * Implement the necessary forms and input fields for user interaction, such as searching for a destination.
5. Implement the navigation functionality:
   * Handle user input and pass it to the Mapbox API to get the required map data and directions.
   * Display the map, route information, and directions on the appropriate pages.
   * Add features like zooming in and out, finding the user's current location, and displaying the shortest route to the destination.
6. Test and refine:
   * Test your web application thoroughly, including different search scenarios and navigation routes.
   * Gather user feedback and make necessary improvements to enhance the user experience.
   * Refactor your code as needed to improve maintainability and performance.

Remember to consult the documentation for the libraries and frameworks you use to ensure you're following best practices and utilizing the available features effectively.