Nexus - Legion of Engineering Solutions

Submitted by:

(101903302) Divyansh Goyal

(101903392) Piyush Bawa

(101903296) Sarthak Gupta

(101903055) Tanvi

BE Third Year- COE

CPG No. 196

Under the Mentorship of

Dr. Seema Bawa

Professor, Computer Science and Engineering

Computer Science and Engineering Department

Thapar Institute of Engineering and Technology, Patiala

Jan-Dec 2022

TABLE OF CONTENTS

Product Perspective

Nexus is a new self-contained product. The website offers a plethora of resources like a roadmap for preparation, course sites, YouTube channels for relevant skills, movies and music, etc. The main functions of this website would be carried out on a remote server system. The concept has been developed with the objective being that it should be able to run on systems comprising older hardware and software and therefore is backward compatible. In addition to this, the product would ensure the secrecy of the information provided by students, society heads, and faculty members. Nexus will introduce students to a roadmap plus a legion of resources to realize the milestones as they go.

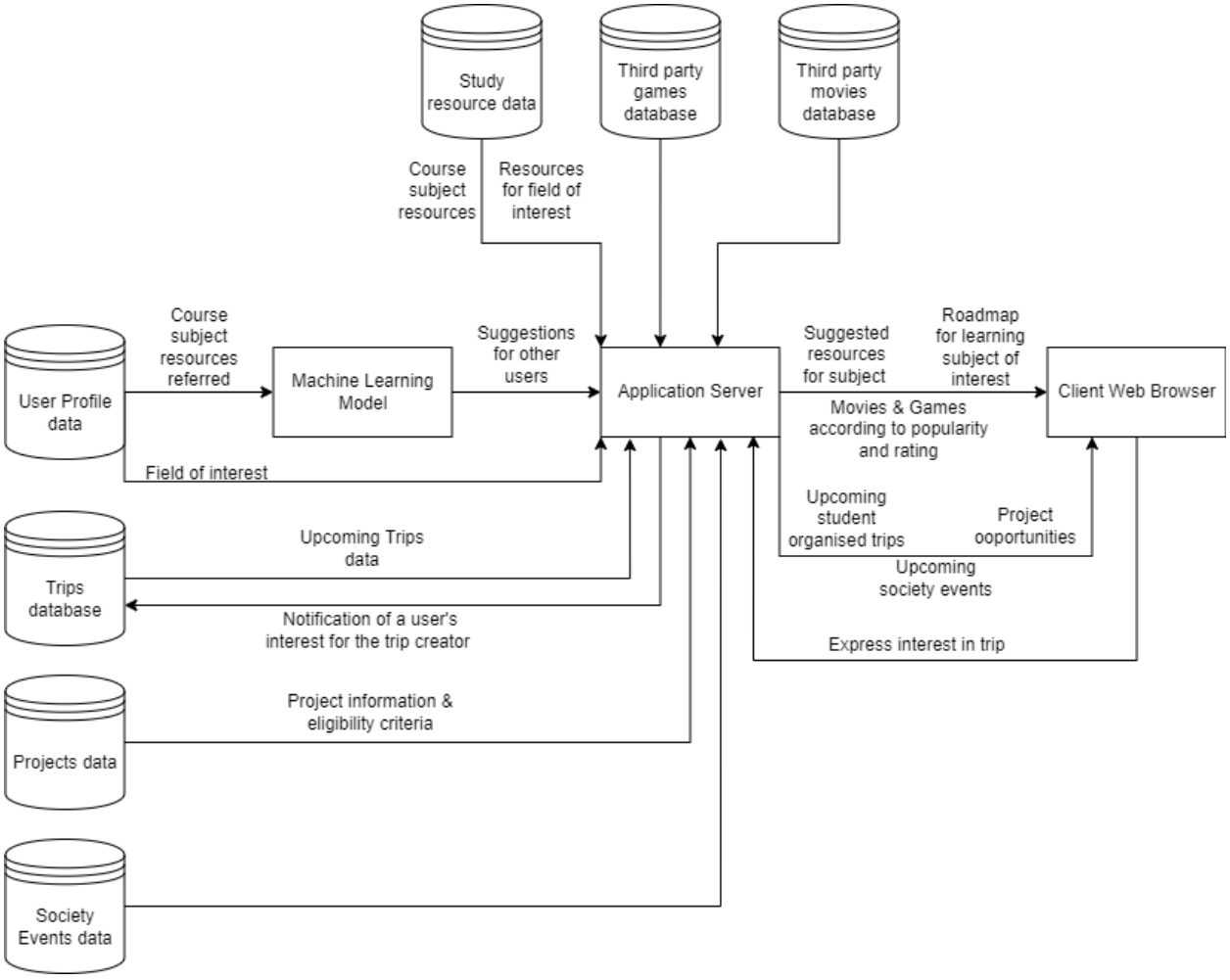


Fig 1. Block Diagram

Use Case Diagram and Templates

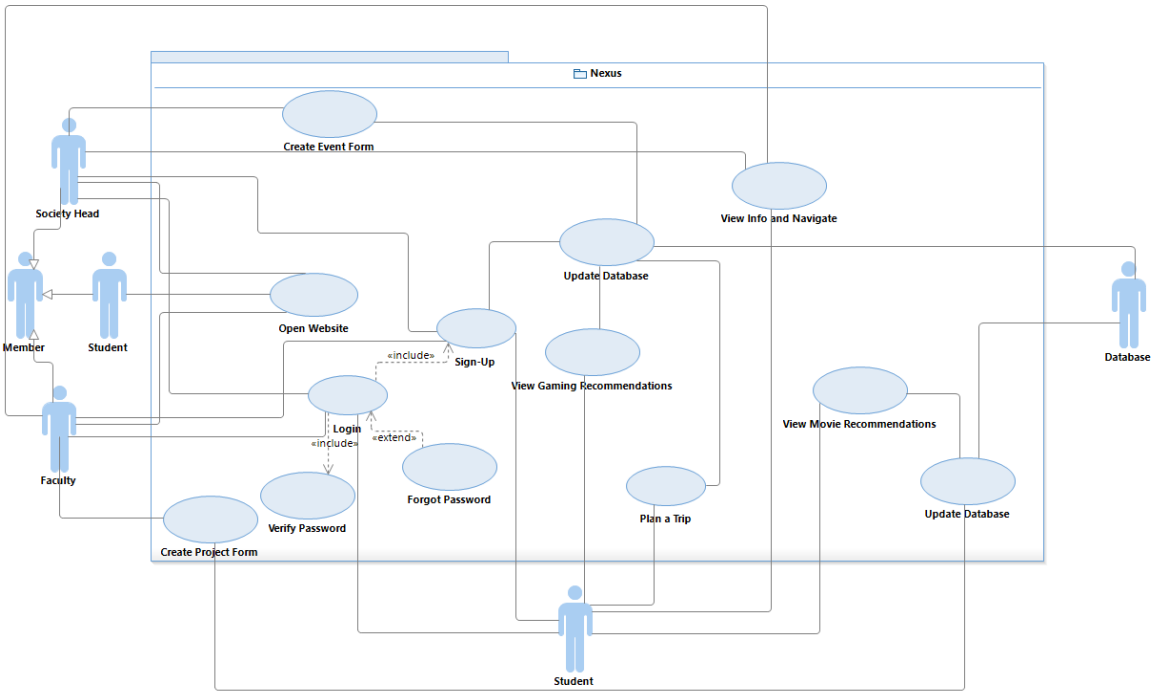


Fig 2. Use Case Diagram

1. Use Case ID	1
2. Use Case Title	Open Website
3. Actors	Faculty, student, society head
4. Purpose: First page displayed when the user visits the website.	
5. Description: Displays general information that doesn't require authentication.	
6. Pre-conditions: 1. Internet access	
7. Task Sequence: 1. The user needs to search 'Nexus' on the browser and open the right link.	
8. Alternate flow: 1. No alternate scenario.	
9. Exceptional flow: 1. The user opens the wrong link while browsing.	
10. Post-conditions: 1. The user logs into the Nexus account. 2. The user scrolls down to check out the upcoming trips, potential projects and events being held.	
Modification History: 11-May-2022	
Author: Piyush Bawa, Divyansh Goyal, Sarthak Gupta, Tanvi	

1. Use Case ID	2
2. Use Case Title	Sign-Up
3. Actors	Faculty, student, society head
4. Purpose: Registration of user accounts on Nexus.	
5. Description: Used for Registration of users on Nexus.	
6. Pre-conditions: 1. The user account should not exist.	
7. Task Sequence: 1. The user visits the website. 2. The user enters the required credentials to create the account.	
8. Alternate flow: 1. No alternate scenario.	
9. Exceptional flow: 1. If the user skips a field, then the field will be highlighted to prompt the user to enter a value.	
10. Post-conditions: 1. The users logs in. 2. The user navigates to various pages of interest.	
Modification History: 11-May-2022	
Author: Piyush Bawa, Divyansh Goyal, Sarthak Gupta, Tanvi	

1. Use Case ID	3
2. Use Case Title	Login
3. Actors	Faculty, student, society head
4. Purpose: For user authentication.	
5. Description: Used for logging into Nexus to avail authenticated features.	
6. Pre-conditions: 1. User account should exist.	
7. Task Sequence: 1. The user needs to enter the credentials and click on submit. 2. The user will be granted access to authenticated services.	
8. Alternate flow: The user has a change of mind and hits “cancel”.	
9. Exceptional flow: If the user enters incorrect login credentials, the system will issue an error to prompt the user to re-enter the details.	
10. Post-conditions: 1. Login credentials are verified. 2. Login error is displayed if details don’t match. 3. User can navigate to various pages and explore features offered on the website.	
Modification History: 11-May-2022	
Author: Piyush Bawa, Divyansh Goyal, Sarthak Gupta, Tanvi	

1. Use Case ID	4
2. Use Case Title	Forgot Password
3. Actors	Faculty, student, society head
4. Purpose: If the user forgets the password.	
5. Description: This feature provides the users to change their password in case they forgot.	
6. Pre-conditions: 1. Login failed 2. The user account should exist.	
7. Task Sequence: 1. User forgets the password. 2. The user clicks on the ‘Forgot Password’ button.	
8. Alternate flow: 1. The user is able to retrieve the previous password.	
Exceptional flow: The user enters the original password.	
9. Post-conditions: 1. The user resets the password.	
Modification History: 11-May-2022	
Author: Piyush Bawa, Divyansh Goyal, Sarthak Gupta, Tanvi	

1. Use Case ID	5
2. Use Case Title	Create Event Form
3. Actors	Society head
4. Purpose: To allow the general secretary of a society/student chapter to update the details of their upcoming event.	
5. Description: This feature allows the users to view upcoming society events.	
6. Pre-conditions: 1. The user must have an account on the website. 2. The user should be logged in.	
7. Task sequence: 1. The user logs into the account. 2. The user enters the event details such as its name, venue, itinerary, prizes etc. 3. The user clicks on 'submit' and the form is created.	
8. Alternate flow: The user has a change of mind and leaves the site.	
9. Exceptional flow: The user misses mentioning some information considered important by the application, an error would be raised.	
10. Post-conditions: 1. A request will be made to the server to store the event details provided by the user. 2. The society head logs out. 3. Students can register for the event(s) of their interest.	
Modification History: 11-May-2022	
Author: Piyush Bawa, Divyansh Goyal, Sarthak Gupta, Tanvi	

1. Use Case ID	6
2. Use Case Title	Create Project Form
3. Actors	Faculty
4. Purpose: To allow a professor to enter the details of a project they are undertaking.	
5. Description: This feature allows faculty to enter the details of a project they are undertaking and for which they need student volunteers to work on.	
6. Pre-conditions: 1. The user must have an account on the website. 2. The user should be logged in.	
7. Task Sequence: 1. The user logs in. 2. The user types in various details regarding the project including the number of positions open and the skills required.	
8. Alternate flow: The user has a change of mind and leaves the site.	
9. Exceptional flow: If the Professor misses mentioning some information considered important by the application, an error would be raised.	
10. Post-Conditions: 1. A request will be made to the server to store the project details provided by the user. 2. The user logs out. 3. Students can apply for the project(s) of their interest.	
Modification History: 11-May-2022	
Author: Piyush Bawa, Divyansh Goyal, Sarthak Gupta, Tanvi	

1. Use Case ID	7
2. Use Case Title	View info and navigate
3. Actors	Faculty, student and society head
4. Purpose: To allow the user to navigate to different pages and display other information.	
5. Description: This is the screen the user would be shown upon logging in and would display essential information apart from allowing the user to navigate to pages of interest.	
6. Pre-Conditions: 1. The user should be logged in.	
7. Task Sequence: 1. The user logs in. 2. The user is directed to dashboard. 3. The user is displayed a complete roadmap related to their interest. The user is shown the resources relevant to the specific interest. 4. The user navigates to the page of interest.	
8. Alternate flow: The user logs out.	
9. Exceptional flow: The user won't be allowed the access to dashboard if the login credentials are incorrect.	
10. Post-conditions: 1. The user is displayed a complete roadmap related to respective interests chosen while registration. The user is shown the resources relevant to the specific interest. 2. The user navigates to the page of interest. 3. The user logs out.	
Modification History: 11-May-2022	
Author: Piyush Bawa, Divyansh Goyal, Sarthak Gupta, Tanvi	

1. Use Case ID	8
2. Use Case Title	View Movie Recommendations
3. Actors	Student, society head
4. Purpose: To show movie recommendations based on peer likes in the website database.	
5. Description: This use case would show the user recommendations for movies according to what their peers liked. They would also be able to click on a particular movie to read about the movie.	
6. Pre-Conditions: 1. The user should have an account on the website. 2. The user should be logged in.	
7. Task Sequence: 1. The user logs in. 2. The user is directed to the dashboard. 3. The user navigates to the movie recommendation page. 4. The user clicks on a movie to read ratings and reviews of the movie.	
8. Alternate flow: 1. The user logs out. 2. The user logs in and navigates to some other page.	
9. Exceptional flow: 1. Server time out due to user traffic.	
10. Post-conditions: 1. The user can navigate to other pages. 2. The user logs out.	
Modification History: 11-May-2022	
Author: Piyush Bawa, Divyansh Goyal, Sarthak Gupta, Tanvi	

1. Use Case ID	9
2. Use Case Title	View Gaming Recommendations
3. Actors	Student, society head
4. Purpose: To show user recommendations for games rated based on peer likes.	
5. Description: This screen would show the user recommendations for games according to what their peers liked.	
6. Pre-Conditions: 1. The user should have an account on the website. 2. The user should be logged in.	
7. Task Sequence: 1. The user logs in. 2. The user is directed to the dashboard. 3. The user navigates to the gaming recommendation page. 4. The user clicks on a game to read about it.	
8. Alternate flow: 1. The user logs out. 2. The user logs in the navigates to some other page.	
9. Exceptional flow: 1. Server time out due to user traffic.	
10. Post-conditions: 1. The user can navigate to other pages. 2. The user logs out.	
Modification History: 11-May-2022	
Author: Piyush Bawa, Divyansh Goyal, Sarthak Gupta, Tanvi	

1. Use Case ID	10
2. Use Case Title	Plan a Trip
3. Actors	Student
4. Purpose: To allow the user to organize and join trips.	
5. Description: The user creates/organizes a trip and then the trip would be visible to his/her peers who can then approach the user if they are interested in tagging along.	
6. Pre-Conditions: 1. The user should have an account on the website. 2. The user should be logged in.	
7. Task Sequence: 1. The user logs in. 2. The user is directed to the dashboard. 3. The user navigates to the trip page. 4. The user clicks on ‘create trip’ button and creates a new trip. 5. The changes are updated in the database.	
8. Alternate flow: 1. The user logs in. 2. The user presses the ‘interested’ button to join a trip. 3. The organizer of the trip will be updated upon the same through a mail. 4. The user information is updated in the database.	
9. Exceptional flow: 1. NULL	
10. Post-conditions: 1. The user checks out other trips. 2. The user creates other trips. 3. The user navigates to other pages. 4. The user logs out.	
Modification History: 11-May-2022	
Author: Piyush Bawa, Divyansh Goyal, Sarthak Gupta, Tanvi	

Set of Tasks and Subtasks

1. Initiation
    - 1.1. Requirements
    - 1.2. Scope(Complete automation and integration)
    - 1.3. Submit Project Proposal
    - 1.4. Project proposal Signed/Approved
2. Design and development
    - 2.1. Front-end Design
    - 2.2. Database Design
    - 2.3. Back-end Development
    - 2.4. Creating Recommendation Model
    - 2.5. Procure Hardware/Software
3. Testing
    - 3.1. System Quality Testing
    - 3.2. Unit Testing
    - 3.3. System Testing
4. Control
    - 4.1. Project Management
    - 4.2. Project Status Meetings
    - 4.3. Risk Management
    - 4.4. Update Project Management Plan

Swimlane Diagram

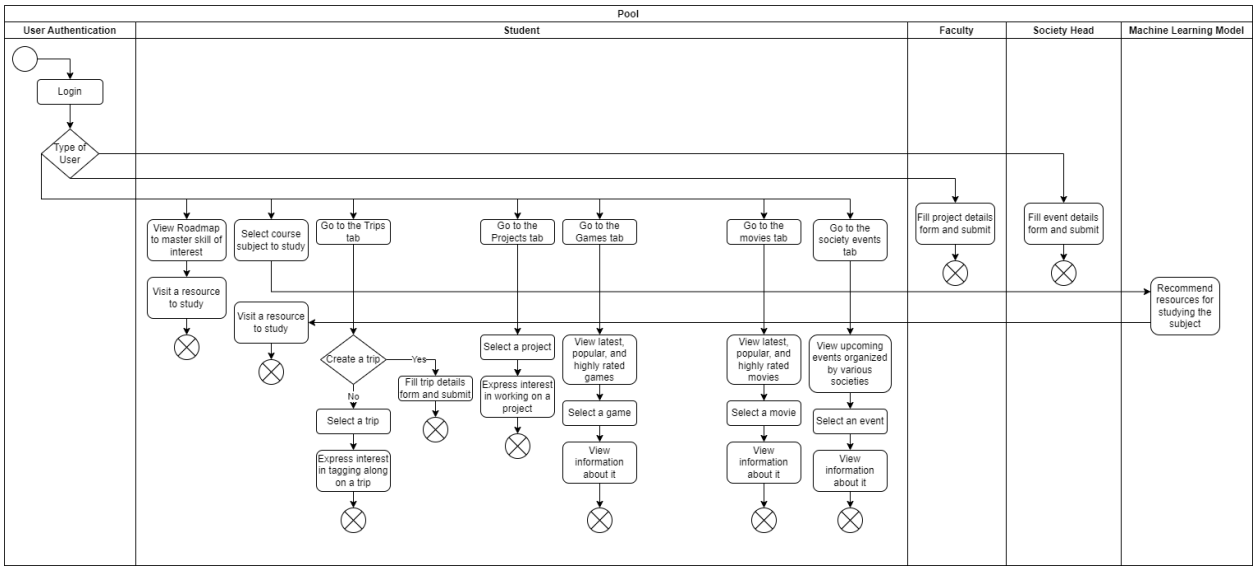


Fig 3. Swimlane Diagram

Work Breakdown Structure

- Outline view
  1. Nexus
     - 1.1. Initiation
       - 1.1.1. Requirements
       - 1.1.2. Scope(Complete automation and integration)
       - 1.1.3. Submit Project Proposal
       - 1.1.4. Project proposal Signed/Approved
     - 1.2. Design and development
       - 1.2.1. Front-end Design
       - 1.2.2. Database Design
       - 1.2.3. Back-end Development
       - 1.2.4. Creating Recommendation Model
       - 1.2.5. Procure Hardware/Software
     - 1.3. Testing
       - 1.3.1. System Quality Testing
       - 1.3.2. Unit Testing
       - 1.3.3. System Testing
     - 1.4. Control
       - 1.4.1. Project Management
       - 1.4.2. Project Status Meetings
       - 1.4.3. Risk Management
       - 1.4.4. Update Project Management Plan

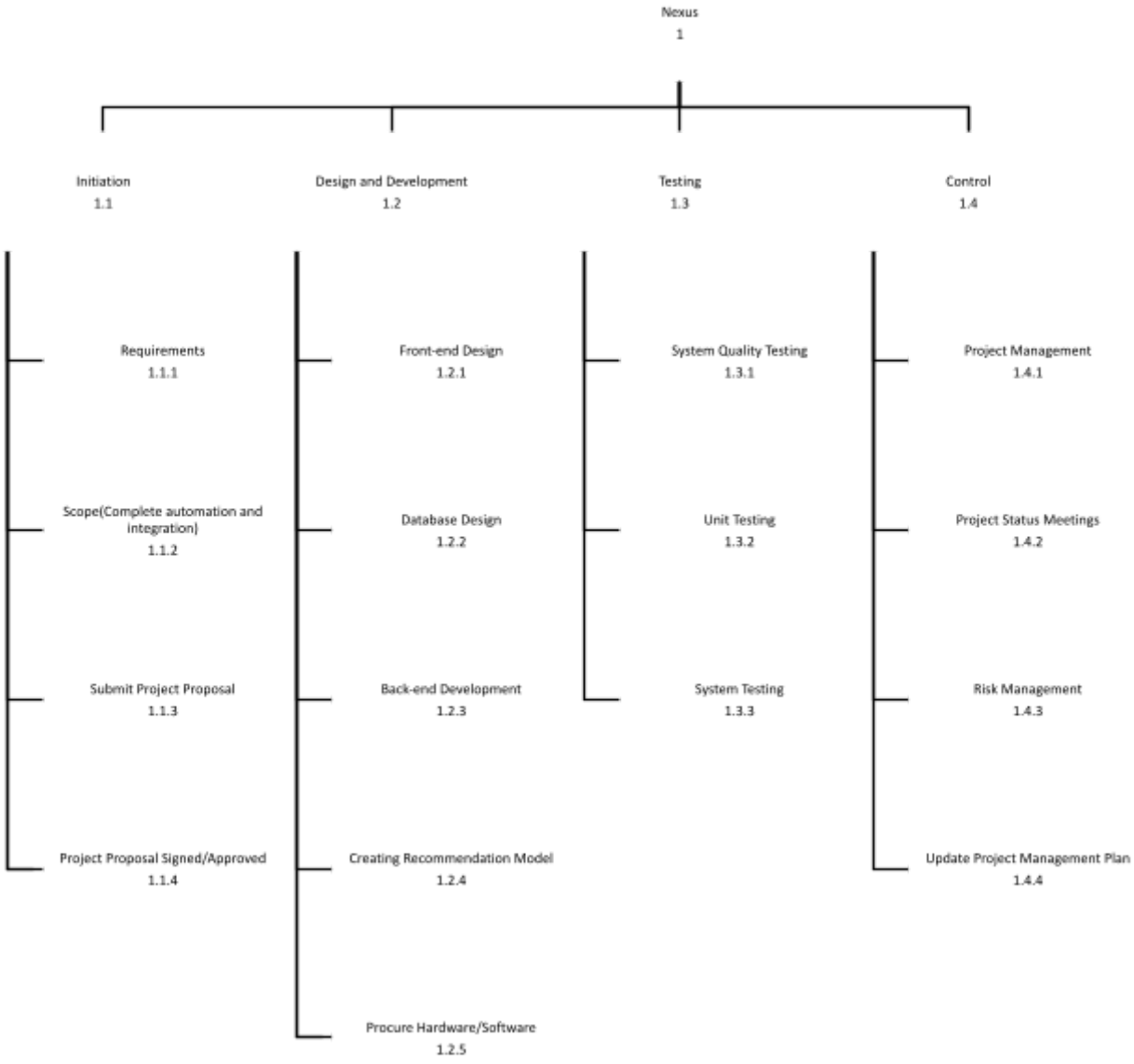
- Hierarchical Structure

Level	WBS Code	Element Name
1	1	Nexus
2	1.1	Initiation
3	1.1.1	Requirements
3	1.1.2	Scope(Complete automation and integration)
3	1.1.3	Submit Project Proposal
3	1.1.4	Project Proposal Signed/Approved
2	1.2	Design and Development
3	1.2.1	Front-end Design
3	1.2.2	Database Design
3	1.2.3	Back-end Development
3	1.2.4	Creating Recommendation Model
3	1.2.5	Procure Hardware/Software
2	1.3	Testing
3	1.3.1	System Quality Testing
3	1.3.2	Unit Testing
3	1.3.3	System Testing
2	1.4	Control
3	1.4.1	Project Management
3	1.4.2	Project Status Meetings
3	1.4.3	Risk Management
3	1.4.4	Update Project Management Plan

- Tabular View

Level 1	Level 2	Level 3
1 Nexus	1.1 Initiation	1.1.1 Requirements 1.1.2 Scope(Complete automation and integration) 1.1.3 Submit Project Proposal 1.1.4 Project Proposal Signed/Approved
	1.2 Design and Development	1.2.1 Front-end Design 1.2.2 Database Design 1.2.3 Back-end Development 1.2.4 Creating Recommendation Model 1.2.5 Procure Hardware/Software
	1.3 Testing	1.3.1 System Quality Testing 1.3.2 Unit Testing 1.3.3 System Testing
	1.4 Control	1.4.1 Project Management 1.4.2 Project Status Meetings 1.4.3 Risk Management 1.4.4 Update Project Management Plan

- Tree Structure View



- WBS Dictionary

Level	WBS Code	Element Name	Definition
1	1	Nexus	All work to implement a system.
2	1.1	Initiation	The work to initiate the project.
3	1.1.1	Requirements	Finalization of the tools and technologies required for the development of the project.
3	1.1.2	Scope(Complete automation and integration)	Discussion of all the activities that are done to develop and deliver the software product.
3	1.1.3	Submit Project Proposal	The team submits the project for approval.
3	1.1.4	Project Proposal Signed/Approved	The mentor signs the project which authorizes the team to proceed further.
2	1.2	Design and Development	Complete project execution.
3	1.2.1	Front-end Design	Creation of the graphic design and user interface.
3	1.2.2	Database Design	Creation of the complete database from the previously collected data.
3	1.2.3	Back-end Design	Development of the server side application and everything that communicates between the database and the browser.
3	1.2.4	Creating Recommendation Model	Study and implementation of the recommendation system using collaborative filtering.
3	1.2.5	Procure Hardware/Software	The procurement of all hardware, software and facility needs for the project.
2	1.3	Testing	The system is tested with a select set of users.
3	1.3.1	System Quality Testing	Testing of the system quality to evaluate the compliance of the system with the corresponding requirements.
3	1.3.2	Unit Testing	All units are individually and independently tested for proper operation.
3	1.3.3	System Testing	Testing of the complete and fully integrated software product.
2	1.4	Control	The work involved for the control process of the project.
3	1.4.1	Project Management	Overall project management for the project.
3	1.4.2	Project Status Meetings	Weekly team status meetings.
3	1.4.3	Risk Management	Risk management efforts as defined in the Risk Management Plan.

Gantt Chart

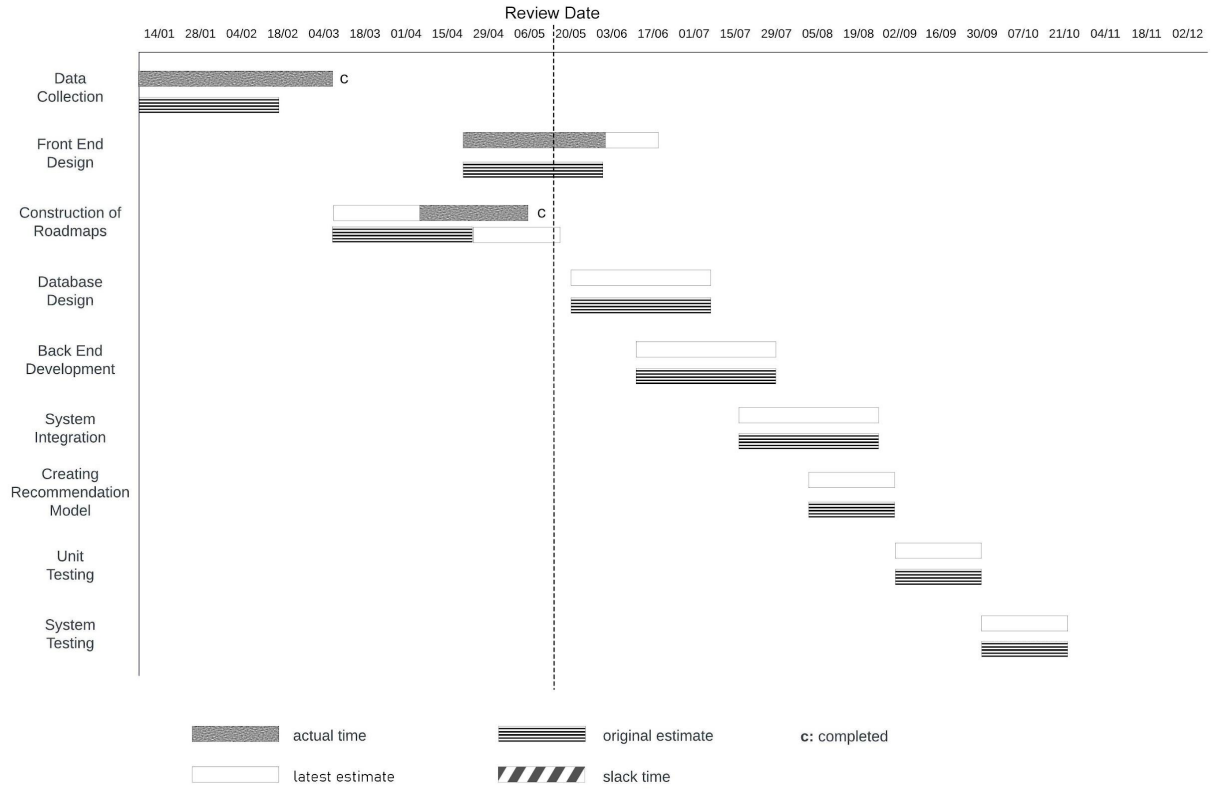


Fig 4. Gantt Chart

Requirements of the Project

1. Functional Requirements

- On the landing page, users should be able to see the information regarding the trips their fellow students are planning, opportunities to work on projects undertaken by professors, and the upcoming events organized by various societies in the university.
- Only authenticated professors should be allowed to enter details of a project they are undertaking and for which they need volunteers.
- Only the general secretary and the society heads should be allowed to enter the details of their upcoming events.
- Users must be allowed to create a trip after logging in.
- Only the authenticated students should be able to access the dashboard.
- The students' dashboard should show the roadmap according to their interests.
- Course names should be listed out to the authenticated users.
- Authenticated users should also be allowed to access the movies, music, and games recommender.

2. Nonfunctional Requirements

2.1 Performance Requirements

Performance requirements define acceptable response times for system functionality. Although the website is developed with the objective being that it should be able to run on systems comprising older hardware and software, the performance will highly depend on the performance of the hardware and software components of the computer, along with the internet connection of the user. When considering the timing relationships of the system in a suitable environment for the platform, the load time for user interface screens shall take no longer than a few seconds. The log-in information shall be verified within a few seconds, and likewise for all other functionalities.

The versions recommended for the dependencies and the system to enhance the user experience are as follows:

- Django: version 3.5 and above (recommended: version 3.8)
- Python: version 2.7.13 and above
- React: React 18.0.1
- Browser: Chromium-based web browser (recommended: Google Chrome)
- OS: Windows/ Linux/ macOS

2.2 Safety Requirements

The following are the vulnerabilities included with the actions that must be taken to prevent the possible exploits:

In case the user's JWT Authentication token gets stolen, the thief would be able to use the app on the user's behalf. They'd be able to view the user's email id and contact details. This is although easy to prevent if the user uses the website as intended and signs out if and when they lend their device to a suspicious person who could have ill intentions.

2.3 Security Requirements

Our product would be able to ensure the secrecy of the information provided by students, society heads, and faculty members at the time of Sign Up. This level of security will be provided at the backend through various encryption libraries, robust database systems, and various checks to eliminate data breaches.

2.4 Software Quality Attributes

1. Availability: The website shall be available at almost all hours
2. Correctness: Extent to which program satisfies specifications, fulfills user's mission objectives.
3. Efficiency: How few resources and time are required to achieve a particular task through the system.
4. Features & Flexibility: Ability to add new features to the system and handle them conveniently.
5. Maintainability: How easy it is to keep the system as it is and correct defects by making changes.
6. Integrity: How the system would insecure the information in the system and how it avoids data losses. Referential integrity in database tables and interfaces.
7. Robustness: Strength of the system to handle system functions accurately and maintain the database without facing unexpected failures.

2.5 Business Rules

The system must have at least a Super-User role and a User role defined for accessing and interacting with the system. Additional roles may be defined for the system as long as the business rules for the administrator and user roles are satisfied. At a minimum, the Super-User roles must account for the data explorer, data curator, and administrator account type requirements. At a minimum, the User role must account for the data explorer account-type requirements described.