**Live Project 1 Part 2**

**Document 6- Please prepare a use case diagram, activity diagram and a use case specification document.**

**Use Case Diagram**



**Activity diagram**

**1.**



**2.**



3.



4.



5.



6.



**Use case specification document –**

**1: Student Registration**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| **Use Case Name** | Student Registration |
| **Description** | Students register on the platform by providing personal and academic details. |
| **Actors** | Primary Actor: Student Secondary Actor: Admin |
| **Basic Flow** | 1. Student accesses the registration portal. 2. Fills out the registration form. 3. Submits the form for verification. 4. System validates and stores the data. 5. Student receives a confirmation message. |
| **Alternate Flow** | None |
| **Exceptional Flow** | 1. Invalid or incomplete data prompts error messages. 2. System denies registration until issues are resolved. |
| **Pre-Conditions** | Student has access to the registration portal. |
| **Post-Conditions** | Registration data is successfully stored. |
| **Assumptions** | Internet access is available. |
| **Constraints** | Limited to new students only. |
| **Dependencies** | System must validate and store data correctly. |
| **Inputs and Outputs** | **Input:** Student details. **Output:** Confirmation message. |
| **Business Rules** | Only one account per student. |
| **Miscellaneous Information** | None |

**2: Fees Payment**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| **Use Case Name** | Fees Payment |
| **Description** | Students pay fees via an online platform. |
| **Actors** | Primary Actor: Student Secondary Actor: Admin |
| **Basic Flow** | 1. Student logs into the portal. 2. Views pending fee details. 3. Selects payment method. 4. Completes payment. 5. Receives confirmation. |
| **Alternate Flow** | 1. Student changes payment method during the process. |
| **Exceptional Flow** | 1. Transaction failure prompts retry or exit options. |
| **Pre-Conditions** | Student has a valid account. |
| **Post-Conditions** | Payment status updated in the system. |
| **Assumptions** | Payment gateway is operational. |
| **Constraints** | Payment limited to defined methods. |
| **Dependencies** | Valid and operational payment gateway. |
| **Inputs and Outputs** | **Input:** Payment details. **Output:** Receipt or error message. |
| **Business Rules** | Payments must be recorded immediately. |
| **Miscellaneous Information** | Late fees may apply. |

**3: Attendance Tracking**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| **Use Case Name** | Attendance Tracking |
| **Description** | Students can view their attendance records. |
| **Actors** | Primary Actor: Student Secondary Actor: Staff |
| **Basic Flow** | 1. Student logs into the portal. 2. Navigates to the attendance section. 3. Views attendance data. |
| **Alternate Flow** | None |
| **Exceptional Flow** | System temporarily unavailable. |
| **Pre-Conditions** | Attendance data must be entered by staff. |
| **Post-Conditions** | Attendance records are displayed to the student. |
| **Assumptions** | Staff updates attendance timely. |
| **Constraints** | Accessible only to enrolled students. |
| **Dependencies** | Accurate and updated attendance data. |
| **Inputs and Outputs** | **Input:** None. **Output:** Attendance details. |
| **Business Rules** | Attendance records must be accurate. |
| **Miscellaneous Information** | None |

**4: Access Course Material**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| **Use Case Name** | Access Course Material |
| **Description** | Students access uploaded course materials. |
| **Actors** | Primary Actor: Student Secondary Actor: Staff |
| **Basic Flow** | 1. Student logs into the portal. 2. Selects a course. 3. Downloads/view materials. |
| **Alternate Flow** | None |
| **Exceptional Flow** | Material not available or download error. |
| **Pre-Conditions** | Course materials must be uploaded by staff. |
| **Post-Conditions** | Student views/downloads materials. |
| **Assumptions** | System storage is operational. |
| **Constraints** | Accessible only to enrolled courses. |
| **Dependencies** | Material uploaded by staff. |
| **Inputs and Outputs** | **Input:** None. **Output:** Course materials. |
| **Business Rules** | Materials must be relevant and updated. |
| **Miscellaneous Information** | None |

**Use Case 5: Track Timetable**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| **Use Case Name** | Track Timetable |
| **Description** | Students view their course timetables. |
| **Actors** | Primary Actor: Student Secondary Actor: Admin |
| **Basic Flow** | 1. Student logs into the portal. 2. Navigates to the timetable section. 3. Views or downloads the timetable. |
| **Alternate Flow** | None |
| **Exceptional Flow** | Timetable not uploaded. |
| **Pre-Conditions** | Timetable must be updated by admin. |
| **Post-Conditions** | Timetable displayed to the student. |
| **Assumptions** | Timetable is accurate and updated. |
| **Constraints** | Accessible only to enrolled students. |
| **Dependencies** | Updated timetable in the system. |
| **Inputs and Outputs** | **Input:** None. **Output:** Timetable details. |
| **Business Rules** | Timetables must be updated per semester. |
| **Miscellaneous Information** | None |

**6: Manage Attendance**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| **Use Case Name** | Manage Attendance |
| **Description** | Staff marks and updates attendance records. |
| **Actors** | Primary Actor: Staff Secondary Actor: Admin |
| **Basic Flow** | 1. Staff logs into the portal. 2. Selects a class/session. 3. Marks attendance. 4. Saves attendance records. |
| **Alternate Flow** | Staff edits attendance after saving. |
| **Exceptional Flow** | System fails to save records. |
| **Pre-Conditions** | Staff must have system access. |
| **Post-Conditions** | Attendance records are updated. |
| **Assumptions** | Accurate data is entered. |
| **Constraints** | Access limited to assigned classes. |
| **Dependencies** | System availability. |
| **Inputs and Outputs** | **Input:** Attendance data. **Output:** Attendance records. |
| **Business Rules** | Attendance must be recorded daily. |
| **Miscellaneous Information** | None |

**7: Update Timetable**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| **Use Case Name** | Update Timetable |
| **Description** | Staff updates course timetables. |
| **Actors** | Primary Actor: Staff Secondary Actor: Admin |
| **Basic Flow** | 1. Staff logs into the portal. 2. Edits timetable details. 3. Saves changes. |
| **Alternate Flow** | None |
| **Exceptional Flow** | System fails to save changes. |
| **Pre-Conditions** | Timetable must exist. |
| **Post-Conditions** | Timetable is updated in the system. |
| **Assumptions** | Data entered is accurate. |
| **Constraints** | None |
| **Dependencies** | System availability. |
| **Inputs and Outputs** | **Input:** Timetable data. **Output:** Updated timetable. |
| **Business Rules** | Timetable updates must be timely. |
| **Miscellaneous Information** | None |

**8: Upload Course Material**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| **Use Case Name** | Upload Course Material |
| **Description** | Staff uploads materials for students. |
| **Actors** | Primary Actor: Staff |
| **Basic Flow** | 1. Staff logs into the portal. 2. Navigates to course section. 3. Uploads files. 4. Confirms upload. |
| **Alternate Flow** | None |
| **Exceptional Flow** | Upload error due to file size or system issue. |
| **Pre-Conditions** | System supports file upload. |
| **Post-Conditions** | Materials are available to students. |
| **Assumptions** | Staff uploads relevant materials. |
| **Constraints** | File size limits. |
| **Dependencies** | Storage system functionality. |
| **Inputs and Outputs** | **Input:** Course material files. **Output:** Confirmation of upload. |
| **Business Rules** | Only authorized staff can upload. |
| **Miscellaneous Information** | None |

**Use Case 9: Manage User Roles**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| **Use Case Name** | Manage User Roles |
| **Description** | Admin assigns and updates user roles. |
| **Actors** | Primary Actor: Admin |
| **Basic Flow** | 1. Admin logs into the portal. 2. Selects a user. 3. Assigns or updates roles. 4. Saves changes. |
| **Alternate Flow** | Admin removes a user role. |
| **Exceptional Flow** | System fails to save changes. |
| **Pre-Conditions** | Users exist in the system. |
| **Post-Conditions** | User roles are updated. |
| **Assumptions** | Role hierarchy is predefined. |
| **Constraints** | Only admins can modify roles. |
| **Dependencies** | None |
| **Inputs and Outputs** | **Input:** Role data. **Output:** Updated roles. |
| **Business Rules** | Roles must align with user permissions. |
| **Miscellaneous Information** | None |

**10: Track Inventory**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| **Use Case Name** | Track Inventory |
| **Description** | Admin monitors inventory levels. |
| **Actors** | Primary Actor: Admin |
| **Basic Flow** | 1. Admin logs into the portal. 2. Views inventory dashboard. 3. Checks stock details. |
| **Alternate Flow** | None |
| **Exceptional Flow** | Dashboard fails to load. |
| **Pre-Conditions** | Inventory data must be up-to-date. |
| **Post-Conditions** | Inventory status is reviewed. |
| **Assumptions** | Stock data is accurate. |
| **Constraints** | None |
| **Dependencies** | Inventory module functionality. |
| **Inputs and Outputs** | **Input:** None. **Output:** Inventory details. |
| **Business Rules** | Only admins can access the inventory module. |
| **Miscellaneous Information** | None |

**11: Schedule and Manage Events**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| **Use Case Name** | Schedule and Manage Events |
| **Description** | Admin creates and manages events. |
| **Actors** | Primary Actor: Admin |
| **Basic Flow** | 1. Admin logs into the portal. 2. Creates/updates event details. 3. Saves event data. |
| **Alternate Flow** | Admin cancels an event. |
| **Exceptional Flow** | System fails to save event data. |
| **Pre-Conditions** | Event data must be complete. |
| **Post-Conditions** | Events are updated in the system. |
| **Assumptions** | Event details are accurate. |
| **Constraints** | None |
| **Dependencies** | System availability. |
| **Inputs and Outputs** | **Input:** Event details. **Output:** Updated events. |
| **Business Rules** | Events must not overlap in schedule. |
| **Miscellaneous Information** | None |

**12: Manage Placements**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| **Use Case Name** | Manage Placements |
| **Description** | Placement officer tracks student placements. |
| **Actors** | Primary Actor: Placement Officer |
| **Basic Flow** | 1. Officer logs into the portal. 2. Updates placement records. 3. Saves data. |
| **Alternate Flow** | None |
| **Exceptional Flow** | System fails to save placement data. |
| **Pre-Conditions** | Placement data is provided. |
| **Post-Conditions** | Placement records are updated. |
| **Assumptions** | Data is accurate and complete. |
| **Constraints** | None |
| **Dependencies** | System availability. |
| **Inputs and Outputs** | **Input:** Placement details. **Output:** Updated placement records. |
| **Business Rules** | Placements must be linked to departments. |
| **Miscellaneous Information** | None |

**13: Coordinate with Companies**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| **Use Case Name** | Coordinate with Companies |
| **Description** | Placement officer interacts with companies for recruitment. |
| **Actors** | Primary Actor: Placement Officer |
| **Basic Flow** | 1. Officer logs into the portal. 2. Updates company interaction records. 3. Saves interaction data. |
| **Alternate Flow** | None |
| **Exceptional Flow** | System fails to save data. |
| **Pre-Conditions** | Company details must be provided. |
| **Post-Conditions** | Interaction records are updated. |
| **Assumptions** | Data is accurate. |
| **Constraints** | None |
| **Dependencies** | None |
| **Inputs and Outputs** | **Input:** Interaction details. **Output:** Updated records. |
| **Business Rules** | Records must be linked to companies. |
| **Miscellaneous Information** | None |

**14: Register on Alumni Platform**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| **Use Case Name** | Register on Alumni Platform |
| **Description** | Alumni register on the platform. |
| **Actors** | Primary Actor: Alumni |
| **Basic Flow** | 1. Alumni accesses the portal. 2. Completes registration form. 3. Submits the form. |
| **Alternate Flow** | None |
| **Exceptional Flow** | Registration fails due to incomplete details. |
| **Pre-Conditions** | Alumni details are available. |
| **Post-Conditions** | Alumni account created. |
| **Assumptions** | Alumni details are accurate. |
| **Constraints** | Registration limited to verified alumni. |
| **Dependencies** | System availability. |
| **Inputs and Outputs** | **Input:** Alumni details. **Output:** Account confirmation. |
| **Business Rules** | One account per alumni. |
| **Miscellaneous Information** | None |

**15: Participate in Events**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| **Use Case Name** | Participate in Events |
| **Description** | Alumni register for college events. |
| **Actors** | Primary Actor: Alumni |
| **Basic Flow** | 1. Alumni logs into the platform. 2. Selects an event. 3. Registers for participation. |
| **Alternate Flow** | Alumni cancels registration. |
| **Exceptional Flow** | System fails to save registration. |
| **Pre-Conditions** | Event details are available. |
| **Post-Conditions** | Registration is saved. |
| **Assumptions** | Alumni details are valid. |
| **Constraints** | None |
| **Dependencies** | Event module functionality. |
| **Inputs and Outputs** | **Input:** Registration data. **Output:** Confirmation. |
| **Business Rules** | Registration limited to event capacity. |
| **Miscellaneous Information** | None |

**Document 7- Screens and pages**

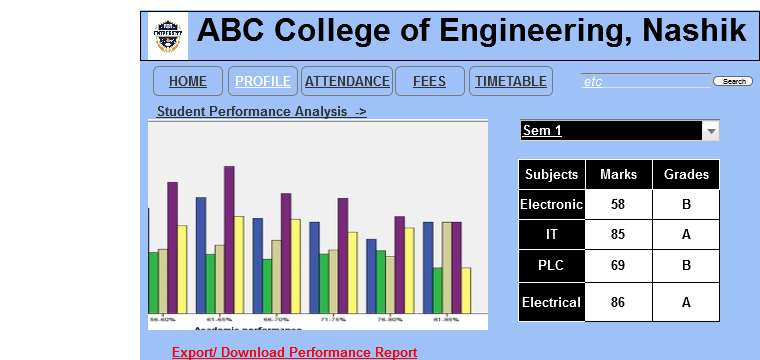
1. Login Page



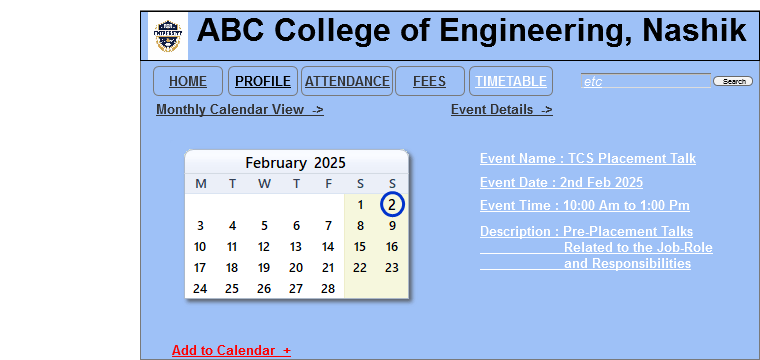
1. Home Page



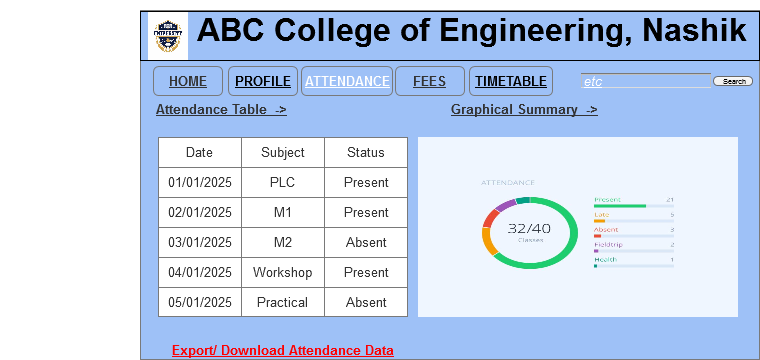
1. Student Profile.



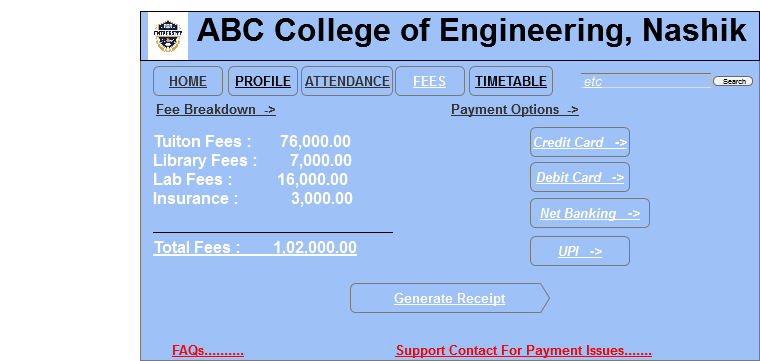
1. Timetable (Event Calendar).



1. Attendance Tracking.



1. Fees Tab.



**Document 8- Tools-Visio and Axure**

Throughout the project, I leveraged Visio and Axure to create visual representations and interactive prototypes of the airline system, enhancing both the design and development phases.

In **Visio**, I utilized its powerful diagramming tools to create detailed flowcharts, activity diagrams, and use case diagrams. The intuitive drag-and-drop interface allowed me to map out complex workflows such as flight bookings, cancellations, and real-time flight status tracking in a clear, structured manner. Visio's ability to connect various shapes and objects helped me maintain a consistent, organized visual structure, ensuring that stakeholders could easily understand the system's processes. Additionally, its integration with Microsoft tools made collaboration seamless, especially when incorporating feedback from team members or during meetings with the client.

On the other hand, I used **Axure** for creating interactive wireframes and high-fidelity prototypes that showcased the user interface (UI) and user experience (UX) of the airline system. Axure's robust functionality allowed me to simulate dynamic interactions, such as flight searches, bookings, and payment flows, helping stakeholders visualize the actual user experience. The ability to add conditional logic and interactions made the prototype highly interactive, providing a realistic representation of how users would navigate the system. This was especially useful during user testing sessions, where real-time feedback could be gathered to iterate on design improvements.

Both tools served distinct but complementary roles in the project. Visio helped in the planning and documentation phases, offering clarity in system workflows, while Axure was crucial for bringing the design to life, allowing stakeholders to interact with the system and providing valuable insights into user behavior and system usability.

**Document 9- BA experience**

**My Experience as a Business Analyst in the College Management System Project:**

**1. Requirement Gathering:**

In this phase, my primary goal was to understand and capture all the requirements for the College Management System accurately and in detail. We used the MOSCOW technique (Must Have, Should Have, Could Have, Won't Have) to prioritize the requirements based on the client's needs. Before starting with MOSCOW, I used elicitation techniques such as interviews, surveys, and workshops to gather all possible inputs from stakeholders.

Since the client was unavailable for a period during this phase, I proactively sourced alternative points of contact from the client’s side, ensuring no communication gaps. I organized regular meetings with these alternate contacts, which allowed me to gather the required information and prevent delays.

Once the initial requirements were gathered, I validated them using the FURPS technique (Functionality, Usability, Reliability, Performance, and Supportability). This helped in ensuring that the requirements were clear, feasible, and realistic. There were instances where duplicate or redundant requirements were identified, so I worked with the stakeholders to remove these duplicates early on to avoid confusion and rework.

During the requirement refinement process, we used prototyping to create visual mockups and simulations, allowing the stakeholders to provide more detailed and specific feedback. The prototypes helped stakeholders better understand their expectations and facilitated a more collaborative approach to refining the requirements.

**2. Requirement Analysis:**

After gathering the requirements, I used UML diagrams to visually represent the system’s structure and components. These diagrams helped in mapping the functional requirements to specific use cases. I also created activity diagrams to visualize the flow of actions, decision points, and interactions between users and the system.

One of the challenges during this phase was dealing with disagreements among the technical team regarding how the system should be structured. I took the initiative to facilitate open discussions and encourage constructive feedback. I also ensured that everyone understood the potential impact of any changes on the project scope, timeline, and cost.

As a result of these discussions, the team reached a consensus, and I was able to update the Business Requirements Specification (BRS) and System Requirements Specification (SRS) documents. These documents outlined the agreed-upon requirements in detail, ensuring that there was no ambiguity moving forward.

**3. Design:**

Once the requirements were analyzed and agreed upon, I moved to the design phase. I created use case diagrams to capture the system’s functional behavior, detailing how users would interact with the system’s components. These diagrams were used to derive test cases, both positive and negative, ensuring that the system was robust enough to handle real-world scenarios. I paid special attention to edge cases during the testing preparation to ensure that the system could handle unusual or unexpected inputs without failure.

While preparing the test cases, I made sure to include negative test cases, which were vital to identify potential system failures, especially in critical areas such as booking, payment, and flight status updates. I also worked on creating the necessary test data and validated that all the test cases were traceable to the specific requirements in the Requirements Traceability Matrix (RTM). This matrix ensured that all the requirements were covered in the design and testing phases.

I also collaborated closely with the technical team and client, ensuring that the design documents accurately represented the business requirements and were aligned with the project’s goals. I took into account feedback from the technical team to ensure that the design was feasible and scalable. This constant communication helped avoid delays in the design phase.

**4. Development:**

During the development phase, I organized and led Joint Application Development (JAD) sessions with both business and technical teams to clarify queries and ensure that the project was on track. These sessions were pivotal in aligning the technical team’s understanding with the business requirements. However, some team members did not initially agree with certain aspects of the design. I managed these situations by holding one-on-one meetings to address their concerns, explaining the rationale behind decisions, and ensuring that they understood the broader impact of the changes.

To maintain team collaboration and avoid disruptions, I encouraged open communication and a healthy work environment. I ensured that the technical team had the necessary documentation and resources to move forward. Regular meetings with the client were held to update them on progress, and I made sure that all team members attended these meetings. In cases where some team members couldn’t attend, I recorded the sessions and shared the recordings with them for further discussions.

**5. Testing:**

In the testing phase, I worked closely with the QA team to prepare test cases from the use cases. The test cases included both functional and non-functional aspects, ensuring comprehensive testing. I also created test data to simulate different use scenarios and validated them against the Requirements Traceability Matrix (RTM) to confirm that all requirements had been covered.

After conducting high-level testing, I helped take client sign-off on the testing phase, ensuring that the system met the client’s requirements and expectations. I also prepared the client for User Acceptance Testing (UAT) by guiding them through the testing process, explaining the necessary steps, and ensuring they were comfortable with the testing environment**.**

**6. Deployment:**

In the deployment phase, I took responsibility for ensuring that all the necessary documentation, including the RTM, was forwarded to the client as part of the project closure. I coordinated with the client to create end-user manuals and training materials, ensuring that users were equipped to operate the system once deployed.

I organized training sessions for end-users to ensure that they understood how to use the system effectively. I also ensured that all stakeholders attended the training sessions and that the training was comprehensive, covering all necessary features. After deployment, I continued to support the project, ensuring that any issues were promptly addressed.

Throughout the project, I acted as the bridge between the business stakeholders, technical team, and the client. I leveraged various techniques such as elicitation, prototyping, JAD sessions, and UML diagrams to ensure that all requirements were clearly defined, validated, and communicated effectively. I maintained strong communication with the client and technical team, ensuring that everyone was aligned with the project’s goals.

As a Business Analyst, I played a vital role in ensuring that the project was completed on time, within scope, and met all of the client’s requirements. My involvement in all phases, from requirement gathering to deployment, allowed me to contribute significantly to the success of the airline system project.