**1.Software Development Methodology:**

**Given the project characteristics, including a new development system, a relatively short time frame (3-6 months), and the expectation of potential changes during development, an Agile methodology would be well-suited for this situation. Agile methodologies, such as Scrum or Kanban, provide flexibility to adapt to changing requirements and deliver incrementally. This approach allows for regular feedback and ensures that the system can be used within 3 months while providing the flexibility to accommodate changes during the development process.**

**2.Testing Strategy:**

**Unit Testing: Developers should perform unit testing to ensure individual components work as expected.**

**Integration Testing: The team should conduct integration testing to verify that different modules work together seamlessly.**

**System Testing: Performed by a dedicated testing team to ensure the entire system functions as intended.**

**User Acceptance Testing (UAT): Involving end-users to validate that the system meets their requirements and expectations.**

**3.Functional and Non-functional Requirements:**

**Functional Requirements:**

**Lecturers/Staffs can register bus routes.**

**Administrative staff can synthesize weekly bus schedules.**

**Admin managers can view bus schedules and approve weekly schedules.**

**Academic staff can transfer lecturers' bus schedules to the administrative department.**

**Non-functional Requirements:**

**Performance: The system must ensure high performance.**

**Security: Users must log in with the FU's email account on the Gmail platform, ensuring information security.**

**4.Use Case Diagram:**

**In this diagram:**

**Actors: Lecturer/Staff and Administrative Staff**

**Use Cases:**

**Register Bus Route (for Lecturer/Staff)**

**Change Bus Route (for Lecturer/Staff)**

**Synthesize Weekly Bus Schedules (for Administrative Staff)**

**View Bus Schedules (for Administrative Staff)**

Lecture/Staff

Administrative Staff

**5.Functional Test Cases:**

***5.1 Register Bus Route:***

**Input: Lecturer/Staff details, route information.**

**Expected Output: Confirmation of successful registration.**

***5.2 Synthesize Weekly Bus Schedules:***

**Input: Weekly schedules for staff and lecturers.**

**Expected Output: Synthesized weekly bus schedules.**

***5.3 View Bus Schedules:***

**Input: Request to view bus schedules.**

**Expected Output: Display of bus schedules.**

***5.4 Approve Weekly Bus Schedules:***

**Input: Approval request from admin manager.**

**Expected Output: Confirmation of approval.**

**6.User Stories:**

**+ As a Lecturer, I want to easily register bus routes, so that I can efficiently plan my transportation to the university.**

**+ As an Admin Manager, I want to quickly approve weekly bus schedules, so that the transportation system runs smoothly.**

**7.Story Map: "Register Bus Route" and "Change Bus Route" Activities:**

**In this story map:**

**Register/Change Bus Route is the main epic.**

**Design, Development, Testing, and Deployment are phases.**

**Specific tasks are listed under each phase, representing the detailed steps for implementing and delivering the feature.**

**This story map provides a clear and organized view of the tasks involved in both registering and changing bus routes for staff/lecturers. It helps in visualizing the workflow and understanding the dependencies between different tasks.**

**Development**

**Design**

**Deployment**

**Testing**

Register/Change Bus Route