βetalink

DESIGN UPDATE DOCUMENT

Ahmed Zaki Al-Gabaly - βeta

βeta-Link PROJECT OWNERSHIP

βeta
AHMED ZAKI AL-GABALY

Taking full ownership of the project and its development activities with a dedicated focus on realizing the βetalink System web application.

- **System Analysis & Design:** Analyzing the Requirements of the system, Designing the System Behaviour & Structure, Planning for the Development of the System.
- **Front End Development:** Ensuring an intuitive, user-friendly interface with a responsive design optimized for various devices and screen sizes.
- **Back End Development:** Developing server-side logic, managing databases, and ensuring secure, seamless communication between the application and its backend systems.

TOC

PROJECT OVERVIEW

Content Overview

This Software Testing Document (STD) outlines the testing strategies, methodologies, and plans for the βetalink Resource Sharing Management Information System for the Faculty of Computing at UTM. It provides a comprehensive guide to ensure the system's quality, reliability, and compliance with defined requirements. The document serves as a reference for testers, developers, and stakeholders to facilitate systematic and efficient testing processes.

The current resource-sharing methods within UTM's Faculty of Computing, primarily relying on email notifications and the default Google Drive interface, have highlighted significant challenges. These include the lack of a structured workflow, inefficient file tracking, and poor search functionalities, which lead to time-consuming retrievals and inconsistencies in resource management. There is a pressing need to improve the user experience by introducing a more organized and efficient resource-sharing solution.

Problem Statement

1. Current resource-sharing methods lack structure, relying on email and the default Google Drive interface.

2. Time-consuming retrieval processes and inconsistencies hinder productivity and organization.

3. Inefficient file tracking and poor search functionalities make resource management difficult.

4. There is a need for a streamlined, user-friendly solution to improve resource sharing and management within the faculty of computing.

Suggested Solution

βeta-Link, improves resource sharing with a structured interface, standardized naming, and advanced search features. It supports role-based access for admins, lecturers, and students. Designed as a Single Page Application (SPA), it ensures seamless access across desktop and mobile platforms, boosting efficiency and productivity.

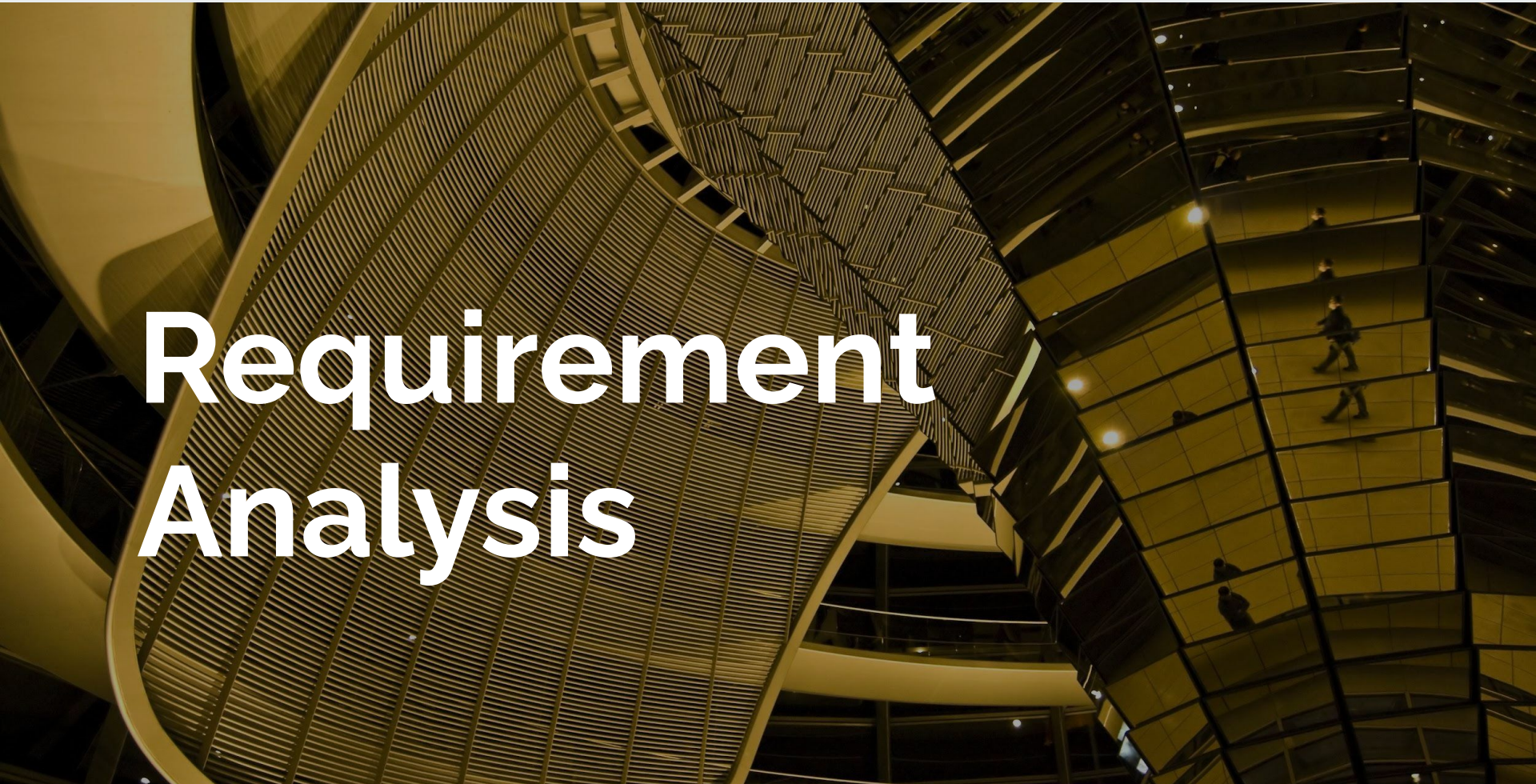
Target audience

UTM requires this system to address inefficiencies in resource sharing, improve accessibility, and ensure proper organization and management of educational materials for students and staff.

01 | **Students**: Need a structured system to easily find and access relevant academic materials.

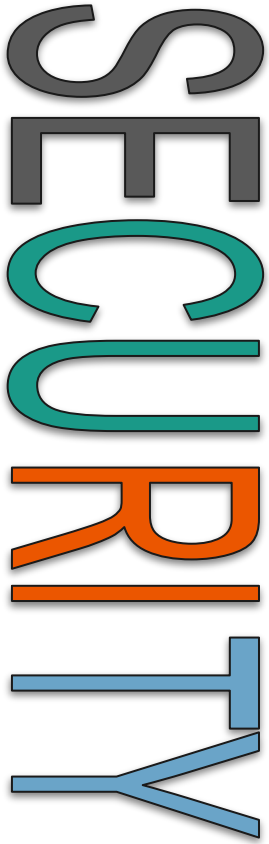
02 | **Lecturers**: Need a tool to upload , organize, and manage educational resources efficiently.

03 | **Faculty Administrators**: Need centralized control to manage Categories organizing the resource distribution.


Requirement
Analysis

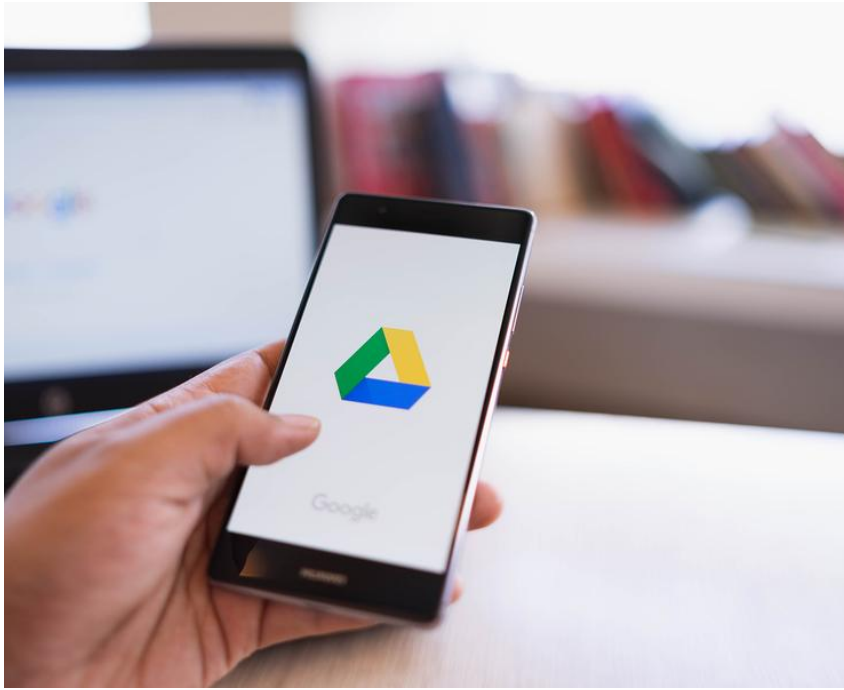
General Requirements

The βetalink platform is designed to improve resource sharing and management for the Faculty of Computing at UTM. The system provides role-based access control, allowing admins, lecturers, and students to perform specific tasks. The platform is a Single Page Application (SPA) built with Vue.js (frontend) and Node.js (backend), using MySQL as the database.



Specific Requirements

- **Admin Requirements**
  - **Manage categories (add, delete).**
- **Lecturer Requirements**
  - **Upload, edit, and delete resources.**
  - **Set resource visibility (visible to students or lecturers only).**
  - **View and search resources.**
- **Student Requirements**
  - **View and search resources shared by lecturers**



Admin Functional Requirements

FR1 (Add Category)	Admins can add new categories for organizing resources.
FR2 (Delete Category)	Admins can delete existing categories.

Lecturer Functional Requirements

FR3 (Upload Resource)	Lecturers can upload resource links with details like name, description, category, and visibility.
FR4 (Edit Resource)	Lecturers can edit resources they have uploaded.
FR5 (Delete Resource)	Lecturers can delete resources they have uploaded.
FR6 (Set Resource Visibility)	Lecturers can set resource visibility (visible to students or lecturers only).
FR7 (View, sort, and Search Resources)	Lecturers can view, sort, and search resources.
FR8 (Auto-Fill)	Lecturer Form to Share a link have some Auto Filled Fields to speed up the process.

Student Functional Requirements

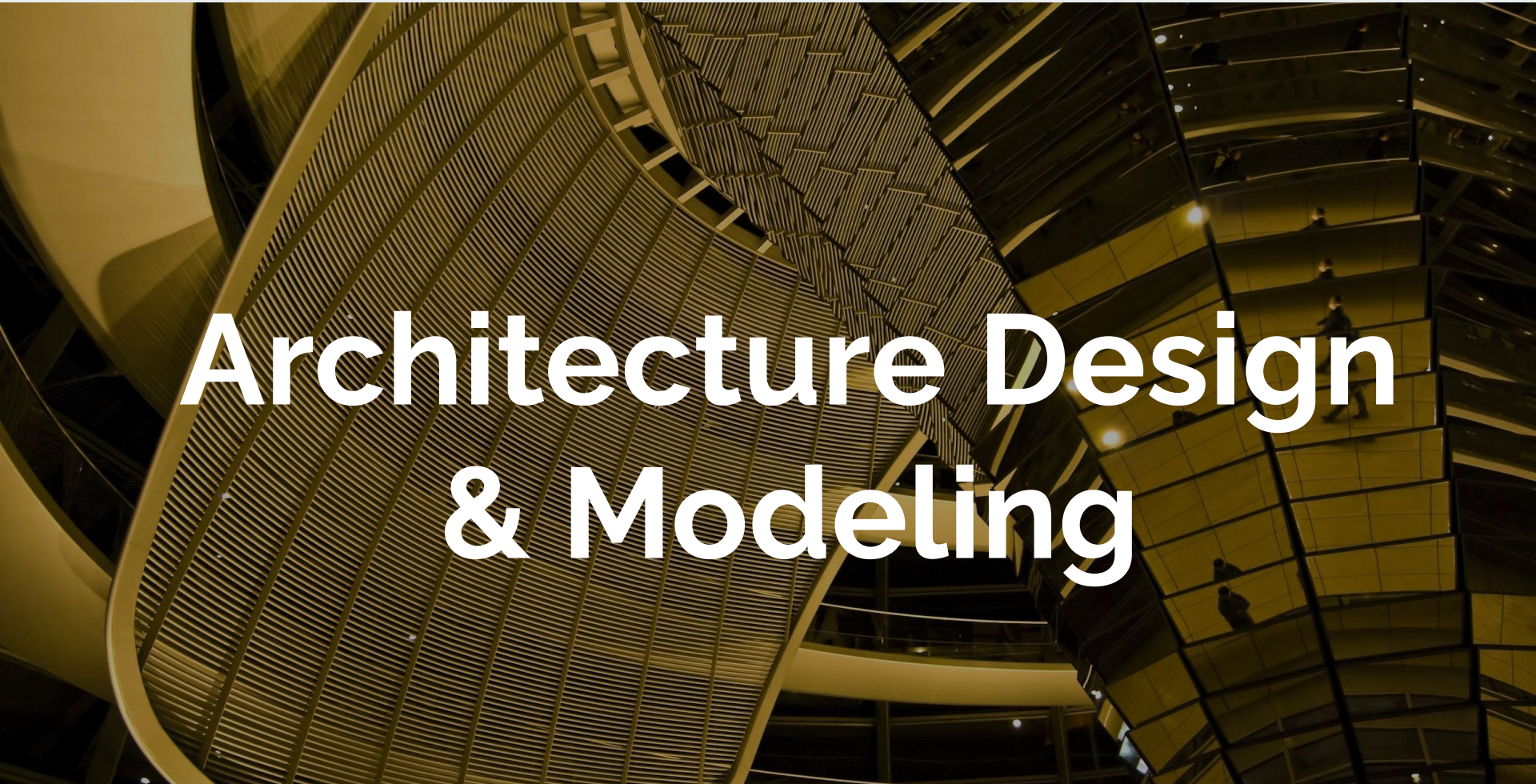
FR9 (View Resources)	Students can view resources shared by lecturers.
FR10 (Search Resources)	Students can search resources by name, category, or description.
FR11 (Delete Resource)	Students can sort resources by 6 different columns.

General Functional Requirements

FR12 (Authentication)	Students, Lecturers, and Admin, All are able to login through the UTM Credentials.
FR13 (View Account)	All user types are able to view some of their UTM Credintals in the Account Page, which are Email and Name.
FR14 (Delete Resource)	Students can sort resources by 6 different columns.

Non- Functional Requirements

NFR1 (Performance)	Resource search results should load within 2 seconds.
NFR2 (Security)	Role-based access control ensures users can only access features permitted for their roles.
NFR3 (Usability)	The user interface should be intuitive and easy to navigate. The platform should have a responsive design, optimized for both desktop and mobile devices.
NFR4 (Scalability)	The system should be scalable to support future growth in the number of users and resources.


Architecture Design
& Modeling

Selected Architecture Style Pattern

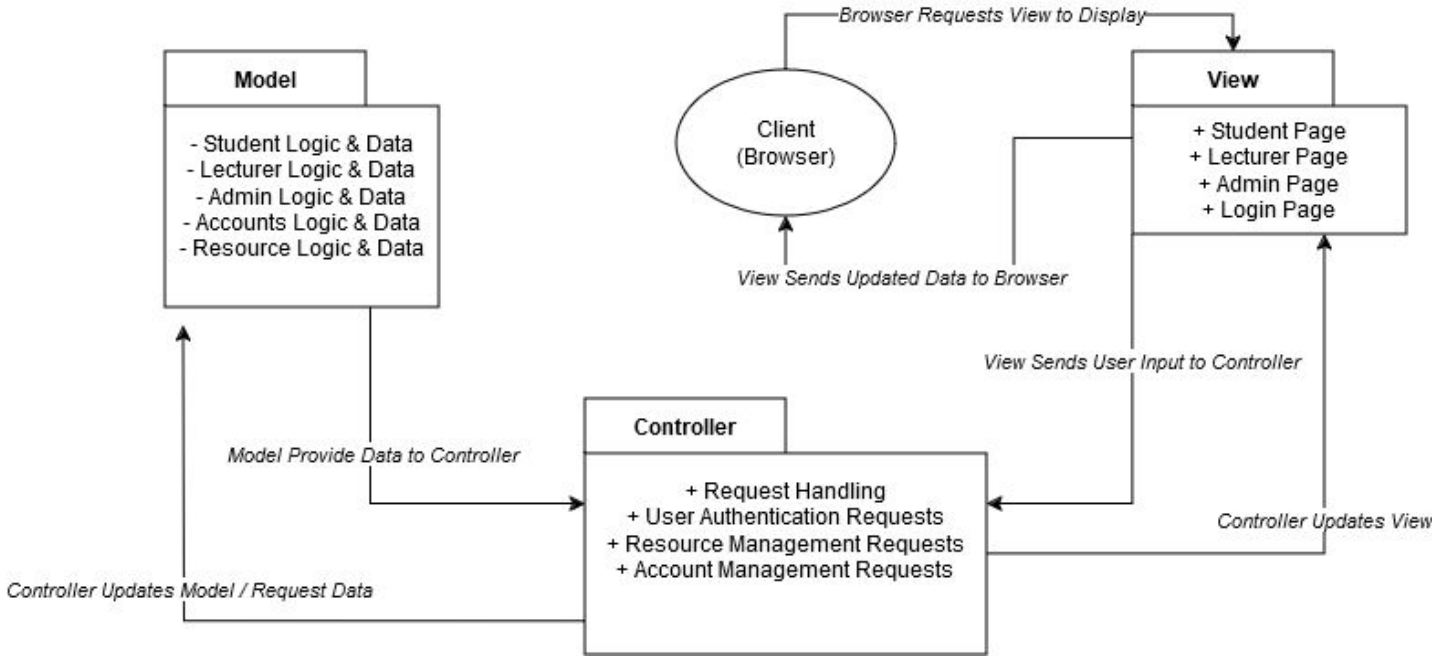
The **Interactive** Architecture Style, specifically the **MVC (Model-View-Controller)** pattern, is the ideal choice for this project:

1. **Separation of Concerns**: MVC separates data, logic, and UI for easier development and maintenance.

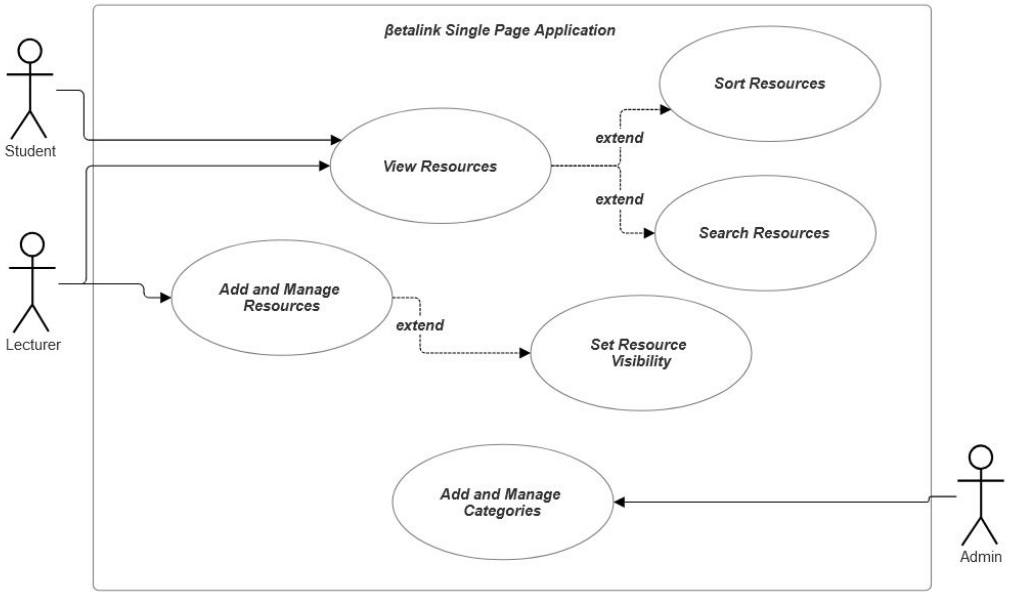
2. **Scalability and Maintainability**: MVC's modular structure allows easy expansion without affecting existing functions.

3. **Role-based Flexibility**: MVC customizes Views and Controllers for different roles while ensuring secure data management.

Architecture Style Diagram (MVC)

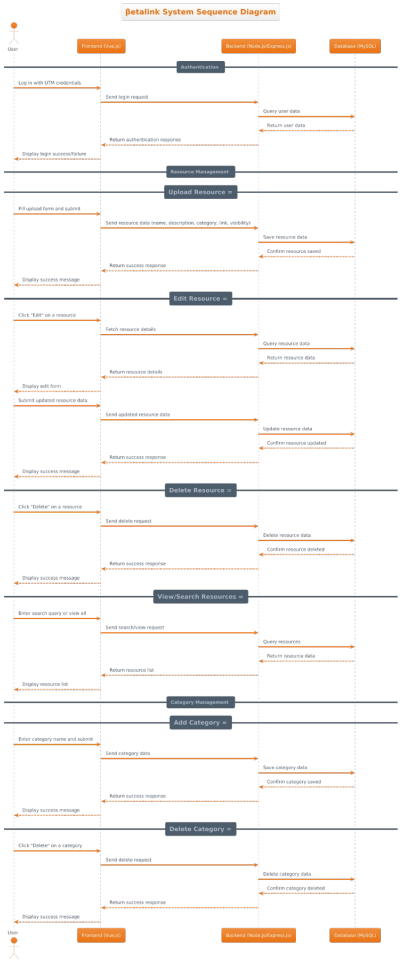


Use Case Diagram



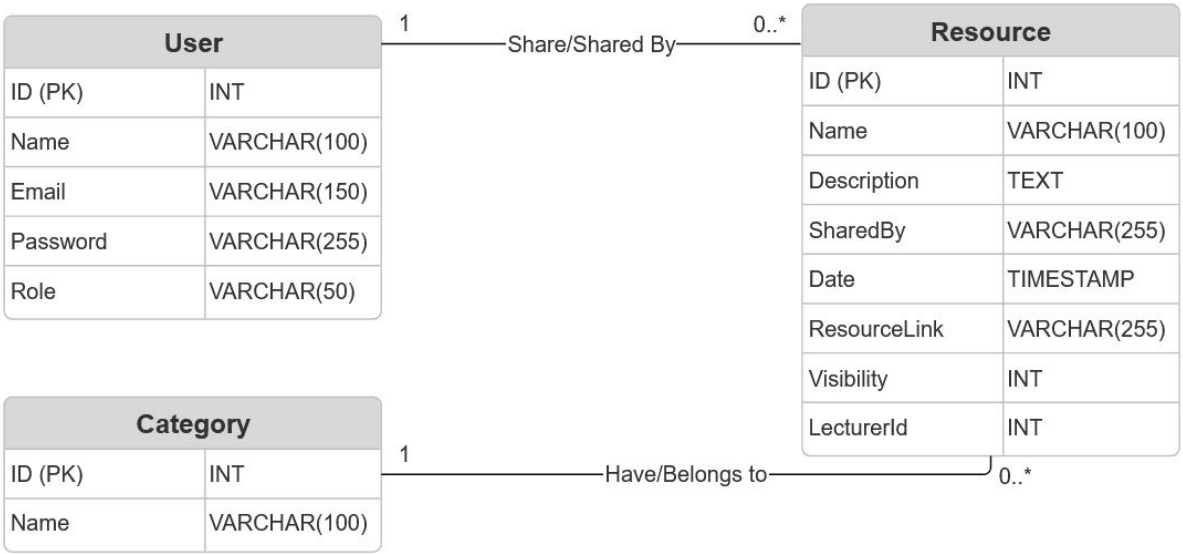
Sequence Diagram

Since the sequence diagram image size is too big to be included in the slides, I will provide it as an image inside the submitted folder, under the name “Sequence Diagram Betalink”




Data Design & User Stories

Entity Relationship Diagram



User Stories (Admin)

As an admin, I want to add a new category so that resources can be organized effectively.

- The admin can enter a category name and click "Add."
- The new category appears in the category list.

As an admin, I want to delete a category so that outdated categories can be removed.

- The admin can click "Delete" next to a category.
- The admin can click "Delete" next to a category.

User Stories (Lecturer)

As a lecturer, I want to upload a resource so that students can access it.

- The lecturer can fill in resource details (name, description, category, link).
- The resource is saved and visible to students (if visibility is set to "Visible to Students and Lecturers").

As a lecturer, I want to edit a resource so that I can update its details.

- The lecturer can click "Edit" next to a resource.
- The lecturer can modify the resource details and save changes.

As a lecturer, I want to delete a resource so that outdated resources can be removed.

- The lecturer can click "Delete" next to a resource.
- The resource is removed from the list.

User Stories (Lecturer)

As a lecturer, I want to set resource visibility so that I can control who can view the resource.

- The lecturer can choose between "Visible to Students and Lecturers" or "Visible to Lecturers Only."
- The visibility setting is saved and applied.

As a lecturer, I want to view and search resources so that I can find relevant materials.

- The lecturer can see a list of resources.
- The lecturer can search resources by name, category, or description.

User Stories (Student)

As a student, I want to view resources so that I can access course materials.

- The student can see a list of resources shared by lecturers.
- The student can click on a resource link to view it.

As a student, I want to search resources so that I can find specific materials.

- The student can enter keywords in the search bar.
- The student can see a filtered list of resources matching the search query.

As a Student, I want to view and sort resources so that I can find relevant materials.

- The student can click on any of the first 6 table columns to sort the resources on the column basis.
- The student can click again on the same column to change the order between ASC to DESC.


TEST PLAN

Scope of Testing

The testing scope focuses on verifying the functionality and usability of the frontend, including navigation across pages, search and sorting in the resource table, category adding and management, resource adding and management, and responsive design across various devices. Backend functionalities like authentication and data persistence are excluded from testing. Tests will be conducted manually using pre-populated JSON data in a simulated environment on standard desktop, laptop, and with modern browsers like Chrome, Firefox, and Edge. IN addition to testing all the functionalities of the system and if it matches the requirements.

Scope of Testing

Milestone	Completion Date	Description
Test Plan Approval	17 Janaury 2024	Finalize and approve this document.
System Tests	21 January 2024	Test Functionalities, views, and other aspects.


TEST CASES

Test Cases

Test Case ID	Description	Objective	Preconditions	Steps	Expected Result	Status
TC1	Navigation	Ensure all navigation links direct to correct pages.	Application is running in a supported browser.	1. Click each navigation link. 2. Verify that the correct content loads for each link.	Each link redirects to the correct page.	PASS
TC2	Resource Table Filtering	Verify search functionality works as intended.	Resource table is populated with data.	1. Enter a keyword in the search bar. 2. Observe if only relevant rows are displayed.	Filtered results display matching rows only.	PASS

Test Cases

Test Case ID	Description	Objective	Preconditions	Steps	Expected Result	Status
TC3	Category Management	Confirm categories can be added and removed.	User is on the category management page.	1. Enter a new category name and click "Add Category." 2. Verify category appears in the list. 3. Click "Delete" to remove a category.	Categories are added and removed successfully.	PASS
TC4	Responsive Design	Verify layout adapts to different screen sizes.	Application is loaded in the browser.	1. Resize browser window or use developer tools to simulate different devices. 2. Observe the layout and functionality.	Layout adjusts without breaking UI elements.	PASS

Test Cases

Test Case ID	Description	Objective	Preconditions	Steps	Expected Result	Status
TC5	Resource Management	Confirm resources can be added and removed.	User is on the Lecturer page.	1. Fill in the resource upload form and submit. 2. Verify resource appears in the list. 3. Fill in resource delete form to remove a resource.	resource are added and removed successfully.	PASS
TC6	Resource Edit	Verify that a lecturer can edit a resource.	Lecturer is logged in, and at least one resource exists.	1. Navigate to the "Resources" page. 2. Click "Edit" next to a resource. 3. Modify the resource details. 4. Click "Save." 5. Verify that the updated	Resource details are successfully updated.	PASS

Test Cases

Test Case ID	Description	Objective	Preconditions	Steps	Expected Result	Status
TC7	Resource Delete	Verify that a lecturer can delete a resource.	Lecturer is logged in, and at least one resource exists.	1. Navigate to the "Resources" page. 2. Click "Delete" next to a resource. 3. Confirm the deletion. 4. Verify that the resource is removed from the table.	Resource is successfully deleted and removed from the table.	PASS
TC8	Resource Visibility	Verify that a lecturer can set resource visibility.	Lecturer is logged in, and at least one resource exists.	1. Navigate to the "Resources" page. 2. Click "Edit" next to a resource. 3. Change the visibility setting (e.g., "Visible to Students and	Resource visibility is successfully updated.	PASS

Test Cases

Test Case ID	Description	Objective	Preconditions	Steps	Expected Result	Status
TC9	Student View Resources	Verify that a student can view resources.	Student is logged in, and at least one resource exists.	1. Navigate to the "Resources" page. 2. Verify that the resource list is displayed. 3. Click on a resource link to view it.	Resource list is displayed, and links are clickable.	PASS
TC10	Student Search Resources	Verify that a student can search for resources.	Student is logged in, and at least one resource exists.	1. Navigate to the "Resources" page. 2. Enter a keyword in the search bar. 3. Verify that only relevant resources are displayed.	Filtered results display matching resources only.	PASS

Test Cases

Test Case ID	Description	Objective	Preconditions	Steps	Expected Result	Status
TC11	Database Connection	Verify that a all CRUD Operation are linked to the Database.	Very Depending on the CRUD Operation	Very Depending on the CRUD Operation	Very Depending on the CRUD Operation	PASS
TC12	Student Sort Resources	Verify that a student can sort resources.	Student is logged in, and at least one resource exists.	1. Navigate to the "Resources" page. 2. Click on of the 6 columns. 3. Verify that resource are sorted on that column basis and clicking again change between ASC and DESC.	Sorted results display data in desired order.	PASS


TEST RESULTS

Summary of the Test Execution

Test Case	Status	Defects
ALL TESTS	PASS	NONE

Deliverables

Design Update (Current Document)

The Current Document.

It outlines the design and architecture of the βeta-Link Resource Sharing Management System for UTM, ensuring alignment with requirements for efficient development and future maintenance.

GUI Using Figma

Link:

UI PROTOTYPE WITH FIGMA

This is a Graphical User Interface Designed using Figma, it consist of 3 page.

Current Progress (Prototype of βetaliink) & Sequence Diagram

The Folders & Source Files Submitted in the E-learning:

BeralinkG.ZIP

Thank you.

Ahmed Zaki Al-Gabaly
βeta