

## Loan User Management Application (LUMA)

### Description:

LUMA provides a facility for GIS Global Employees to purchase items from GIS Global Mart and facilitates a flexible loan to purchase the items. GIS Global issues loan cards for various purchases like furniture, stationary, crockery, etc., to all their employees with specific repayment tenure for each of the loan type. Whenever an employee applies for loan, based on his eligibility loan will be approved and assign an employee with loan card. Each employee can be issued with different types of cards for purchasing different categories of products.

There are two types of users

1. Admin, who is an Internal Manager person manages the data.
2. User, who is an internal Employee who can apply for the card

LUMA, is to be developed to manage the User activities.

The main functionalities of the LUMA are:

1. It should allow the user to login and validate the credentials.
2. It should allow the users to apply loan for a particular item
3. It should allow the users to view all the loan cards available
4. It should allow the users to view all the items purchased on loan cards

### Proposed Wireframes:

- a. A standard login screen to validate user credentials

Loan Management Application

User Login

Enter User id

Enter User Password

b. User dash board to select various operations

Loan Management Application

User Dash board

View Loans

Apply for Loan

View Items Purchased

c. Apply for Loan

Loan Management Application

Select Product and Apply for Loan

Employee id

Item Category

Furniture

Item Description

Item value

Item Make

Wodden

Apply Loan

d. Wireframe to display all Loan cards of particular employee

Loan Management Application

Loan Cards Aailed

Employee id: E100001

Designation: Manager

Department: Marketing

Load id	Loan Type	Duration	Card Issue Date
L00001	Furniture	5	1/1/2000
L00002	Stationary	0	1/1/2000
L00003	Crockery	1	2/2/2002

- e. Wireframe to display all items purchased

Loan Management Application				
Items Purchased				
Employee id: E100001		Designation: Manager		Department: Marketing
Issue_id	Item Description	Item Make	Item Category	item valuation
100001	Tea Table	Wodden	Furniture	5000
100002	Tea Set	Glass	Crockery	2000
100003	Dining Set	Wodden	Furniture	15000
1000004	Pen	Plastic	Statinary	1000

### 3. Toolchain

Databases: MySQL

Presentation or View Layer: React

Backend processing: Spring and Springboot

Database Connectivity: Spring data JPA

Version control systems: Git

Build Tools: Maven

## Development Flow

The application development should be completed in 40 hours, as per the below order

Phase -1 : Backend Development: Backend Tasks – Code Project panel presentation

Phase -2: Frontend Development: Frontend Tasks – Code project panel presentation

## Business Requirement:

There will be 5 main user stories required to be implemented covering the below use cases:

### User Stories

User story Id	Us-01
User story title	User Login
User Story Details	1. User should provide the user id and password for validation.
Acceptance Criteria	1. Both username and password are mandatory, if not provided, error messages should be displayed. 2. Successful validation should redirect to menu page, unsuccessful validation should redirect to registration page.

User story Id	Us-02
User story title	User Menu
User Story Details	Display the User Dashboard for View Loans, Items Purchased, Apply for Loan
Acceptance Criteria	Null

User Story Id	Us_03
User Story Title	Apply Loan
User story Details:	An user should be able to purchase a product from the available list, by providing the appropriate details
Acceptance Criteria	Upon selecting the category from the list, its associated list of products should be populated in the list, once an item is selected its value should be automatically populated in the text box.

User_Story id	Us_04
User story Title	View Loan Cards
User Story Details:	An user should be able to view all the loan cards availed
Acceptance Criteria	It should display only the active loan cards.

User_Story_id	Us_05
User Story Title	View Items Purchased
User Story Details	An user should be able to view all the items purchased
Acceptance Criteria:	It should display all the products which are purchased, both active loans and completed loans.

Note: A separate service component must be created to call the spring boot backend services and all the validations or processing regarding the use case should be done at the backend only.

### Backend Layer User stories

User story Id	Us-01
User story title	User Login processing
User Story Details	<ol style="list-style-type: none"> <li>1. Should be able to extract the values from request body using @RequestBody</li> <li>2. Read the user details from database using spring data jpa and validate it with the UI values.</li> <li>3. After validating should send response to React UI</li> <li>4. Must use GET Method of communication</li> </ol>
Acceptance Criteria	<ol style="list-style-type: none"> <li>1. Once user validation is done, view must return the main menu in react</li> <li>2. If validation fails, view must return to login page only, as this application does not allow for admin to register themselves.</li> <li>3. All validations must be performed at backend only</li> </ol>

User story Id	Us-02
User story title	Loan processing
User Story Details	<ol style="list-style-type: none"> <li>1. Should be able to extract the values from request body object</li> <li>2. Read the loan details from UI and pass it to service layer further to do layer based on item selected, it should identify the appropriate card type and amount, do the processing</li> <li>3. Must use POST method of communication</li> </ol>
Acceptance Criteria	<ol style="list-style-type: none"> <li>1. Once data is inserted, it should redirect to dash board</li> <li>2. If insertion fails, it should display an error page and must have provision to go to menu page.</li> </ol>

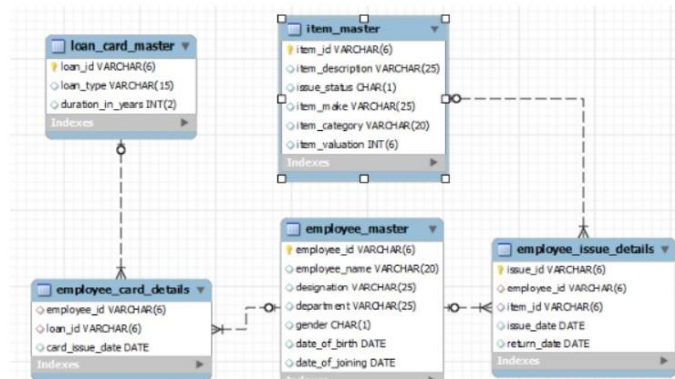
User story Id	Us-03
User story title	View Loans
User Story Details	<ol style="list-style-type: none"> <li>1. Should be able to extract the values from request body object</li> <li>2. Based on the employee id logged in, it should find and display the appropriate active loans</li> <li>3. Must use POST method of communication</li> </ol>
Acceptance Criteria	It must have a provision to redirect to home or dashboard upon successful display

User story Id	Us-04
User story title	View Items Purchased
User Story Details	<ol style="list-style-type: none"> <li>1. Should be able to extract the values from request body object</li> <li>2. Based on the employee id logged in, it should find all the items purchased by the employee and returns them in the form of grid.</li> <li>3. Must use POST method of communication</li> </ol>
Acceptance Criteria	<ol style="list-style-type: none"> <li>1. It must have a provision to redirect to home or dashboard upon successful display.</li> </ol>

#### Database Layer

User story	User Story Details
Us_01	<ol style="list-style-type: none"> <li>1. DB Schema creation and setup in mysql database</li> <li>2. Spring boot project setup creation.</li> <li>3. Develop the post method api to read data from view page.</li> <li>4. Use appropriate DTO objects for view and data integration</li> <li>5. Use spring data jpa for connecting to databases.</li> </ol>
Us_02	<ol style="list-style-type: none"> <li>1. Set up the appropriate methods to perform functions like user validation, user registration, transaction management, transaction details.</li> </ol>

## Database Schema



## Presentation:

- I. No custom CSS, UX framework like bootstrap must be used
- II. An Appropriate GoF design pattern should be implemented to compose and process the data received from backend APIs
- III. SOLID principles should be implemented to develop reusable and modular components
- IV. UI app should have appropriate client-side validations
- V. UI app should have the latest versions of available imported packages and libraries

## Methodology:

Agile-based development methodology should be used to track and manage the progress of the whole process. As a developer, it is expected to update the Agile tools like JIRA with status updates and impediments (Optional)

#### Day wise plan for user stories

Day -1	Database Layer Us_01 , Us_02 , Frontend US_01
Day-2	Backend US_01 and Frontend Us_02
Day-3	Frontend US_03, Frontend US_04, Backend Us_02 and Backend US_03
Day-4	Frontend US_05 Backend Us_04
Day- 5	Unit test cases, testing and ppt preparation.