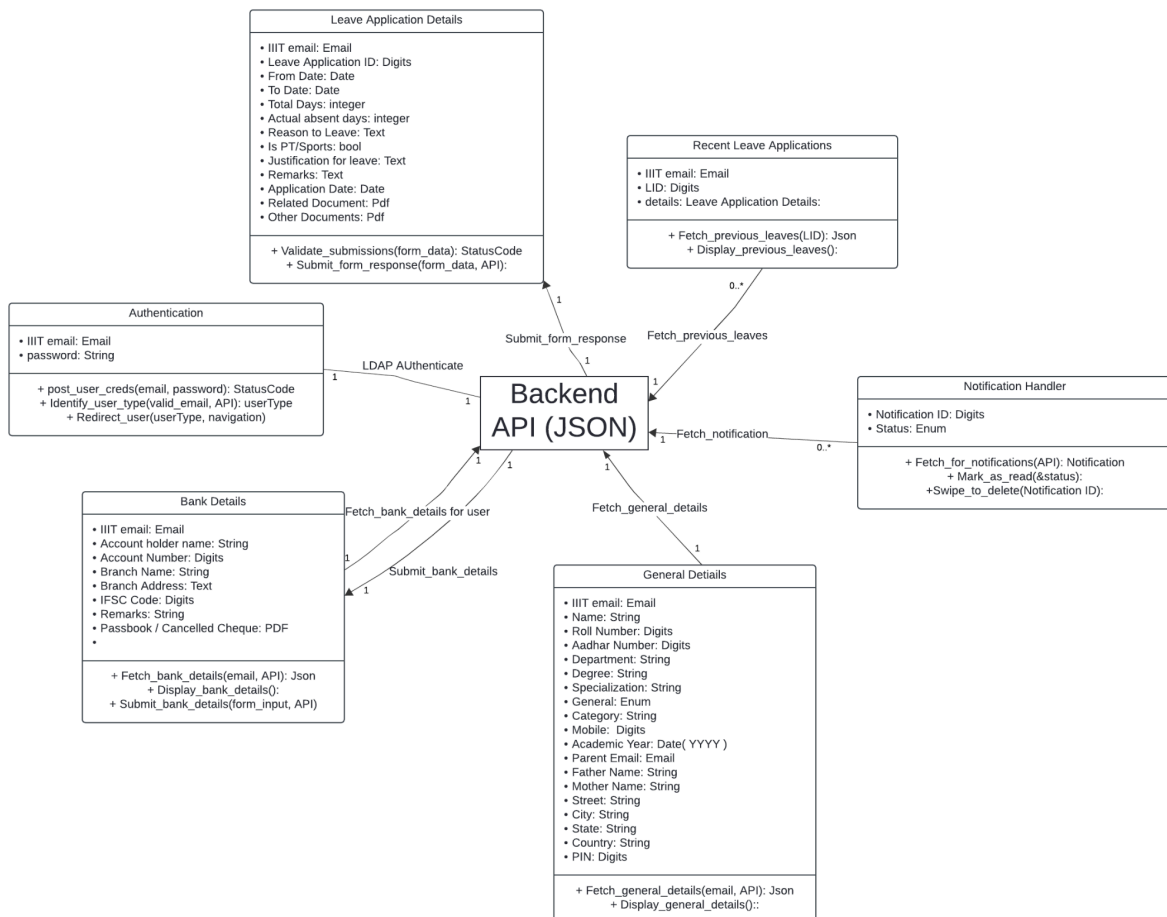


# Product Design

Team-15: Abhiram Tilak, Hemanth Sunkireddy, Jakeer Hussain,  
Samyak Mishra

## Design Model



Link: <https://lucid.app/documents/view/151acb36-7aef-4608-9ed2-1645f5e533e8>

<b>API (provided by client )</b>	<p><b>Class state:</b> Has access to all the storage and login in the IMS backend.</p> <p><b>Class Behaviour:</b></p> <ul style="list-style-type: none"> <li>• Json input ( takes input in json format when app performs POST)</li> <li>• Json output ( gives output in json format when app performs GET)</li> </ul>
----------------------------------	---

<b>Authentication</b>	<p><b>Class state:</b> Contains the information required to perform LDAP authentication using the APII</p> <ul style="list-style-type: none"> <li>• Email-ID (username/uid)</li> <li>• Password</li> </ul> <hr/> <p><b>Class Behaviour:</b></p> <ul style="list-style-type: none"> <li>• post_user_creds(): Send the LDAP API the credentials of username and password.</li> <li>• Identify_user_type(): Used to identify if the valid user ( if any ) is a faculty or student.</li> <li>• Redirect_user(): Redirects to different pages based on action ( either dashboard or forgot password page )</li> </ul>
-----------------------	--

<b>Bank Details</b>	<p><b>Class State:</b> Contains bank details of a valid user.</p> <ul style="list-style-type: none"> <li>• Email (unique-id/username)</li> <li>• Account holder name</li> <li>• Account number</li> <li>• Branch Name</li> <li>• IFSC Code</li> <li>• Branch Address</li> <li>• Remarks</li> <li>• Passbook / Cancelled Cheque</li> </ul> <hr/> <p><b>Class Behaviour:</b></p> <ul style="list-style-type: none"> <li>• Fetch_bank_details(): Used to fetch bank details from API</li> <li>• Display_bank_details(): After fetching displays the details.</li> <li>• Submit_bank_details(): In the edit bank details page when the bank details are submitted the new contents get sent to the server.</li> </ul>
---------------------	---

<b>ProfileDetails</b>	<p><b>Class state:</b> This stores all the personal details of the user</p> <ul style="list-style-type: none"> <li>• Email ( unique-id / username )</li> <li>• Name</li> <li>• Roll Number</li> <li>• Date of Birth</li> <li>• Aadhar Number</li> <li>• Department</li> <li>• Degree</li> <li>• Specialisation</li> <li>• Gender</li> <li>• Category</li> <li>• Mobile</li> <li>• Academic year</li> <li>• Admission time</li> <li>• Parent email</li> <li>• Father Name</li> <li>• Mother Name</li> <li>• Street</li> <li>• City</li> <li>• State</li> <li>• Country</li> <li>• Pin Code</li> </ul> <hr/> <p><b>Class Behaviour:</b></p> <ul style="list-style-type: none"> <li>• Fetch_profile_details(): Used to fetch general details of the user from IMS</li> <li>• Display_profile_details(): After fetching displays the details</li> </ul>
-----------------------	---

<b>Leave Application Details</b>	<p><b>Class State:</b> This is the main class that contains details about the leave application form.</p> <ul style="list-style-type: none"> <li>• LID ( Leave Application ID)</li> <li>• Email ( username/ uid )</li> <li>• From Date</li> <li>• To Date</li> <li>• Total days</li> <li>• Actual absent days</li> <li>• Reason for leave</li> <li>• Is PT/Sports only ( bool )</li> <li>• Justification for leave</li> <li>• Application Date</li> <li>• Related Documents</li> <li>• Other Documents</li> </ul> <hr/> <p><b>Class Behaviour:</b></p> <ul style="list-style-type: none"> <li>• Validate_submission(): This method makes sure all fields are filled and they are of right type</li> <li>• Submit_from_response(): Handles submission of the leave applications to the API.</li> </ul>
----------------------------------	---

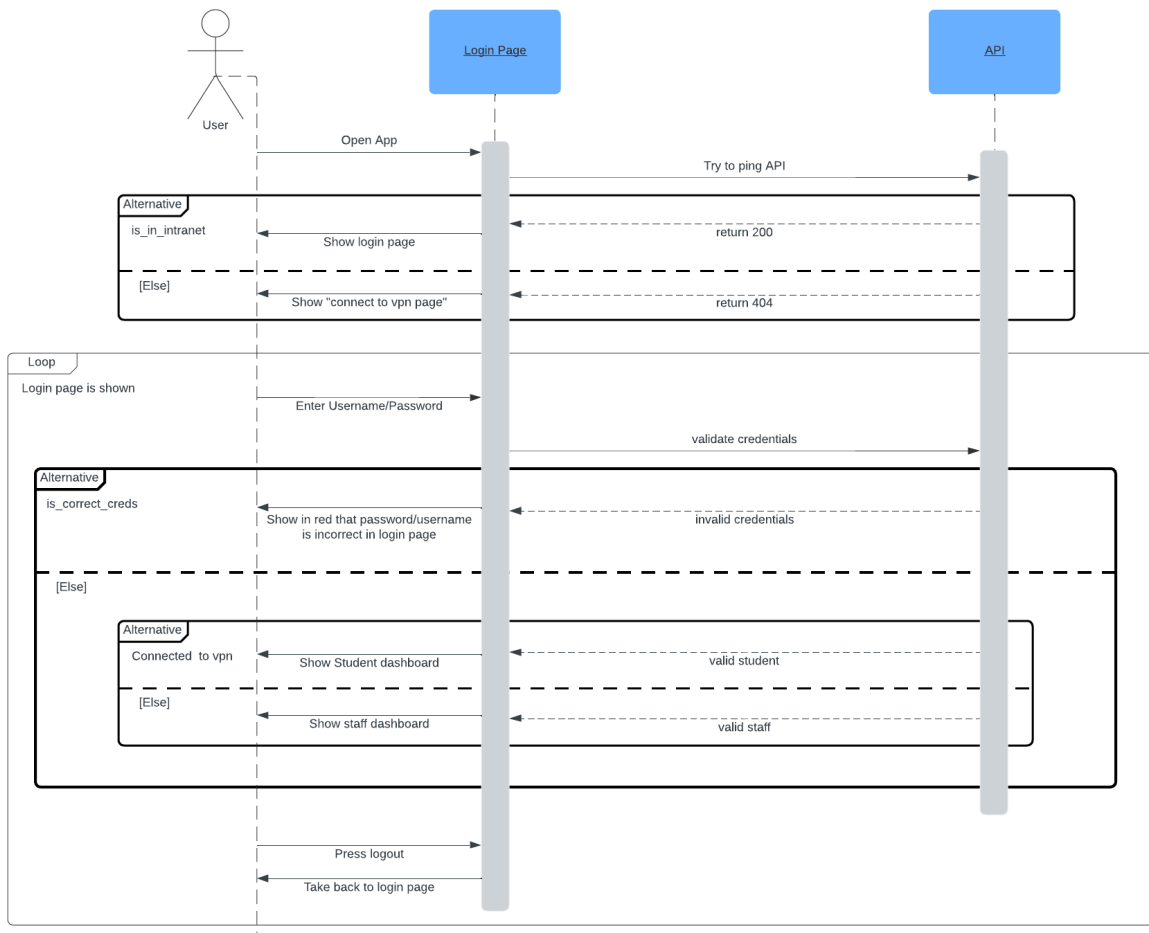
<b>Recent Leave Applications</b>	<p><b>Class State:</b> This class is used to handle previous submissions of leave applications and their status.</p> <ul style="list-style-type: none"> <li>• Email ( username/ uid )</li> <li>• Leave Application ID (LID)</li> <li>• Leave Application Details for the given LID</li> </ul> <hr/> <p><b>Class Behaviour:</b></p> <ul style="list-style-type: none"> <li>• Fetch_previous_leaves(): Used to fetch from the API information about previous submissions of leave applications</li> <li>• Display_previous_leaves(): Display the contents of previous leave applications along with the “status” of that leave.</li> </ul>
----------------------------------	--

<b>Notification Handler</b>	<p><b>Class State:</b> This class is used to handle situations like leave applications getting approved or transcript getting updated.</p> <ul style="list-style-type: none"> <li>• Notification ID</li> <li>• Status: (read / unread )</li> </ul> <hr/> <p><b>Class Behaviour:</b></p> <ul style="list-style-type: none"> <li>• Fetch_for_notifications(): Everytime the notification handler fetches data from API, either an empty list is returned or a list of notifications is returned which are to be displayed.</li> <li>• Mark_as_read(): Once the notification is clicked on, change its status to read from unread.</li> <li>• Swipe_to_delete(): Everytime you swipe right on a notification, it gets deleted.</li> </ul>
-----------------------------	--

# Sequence Diagram(s)

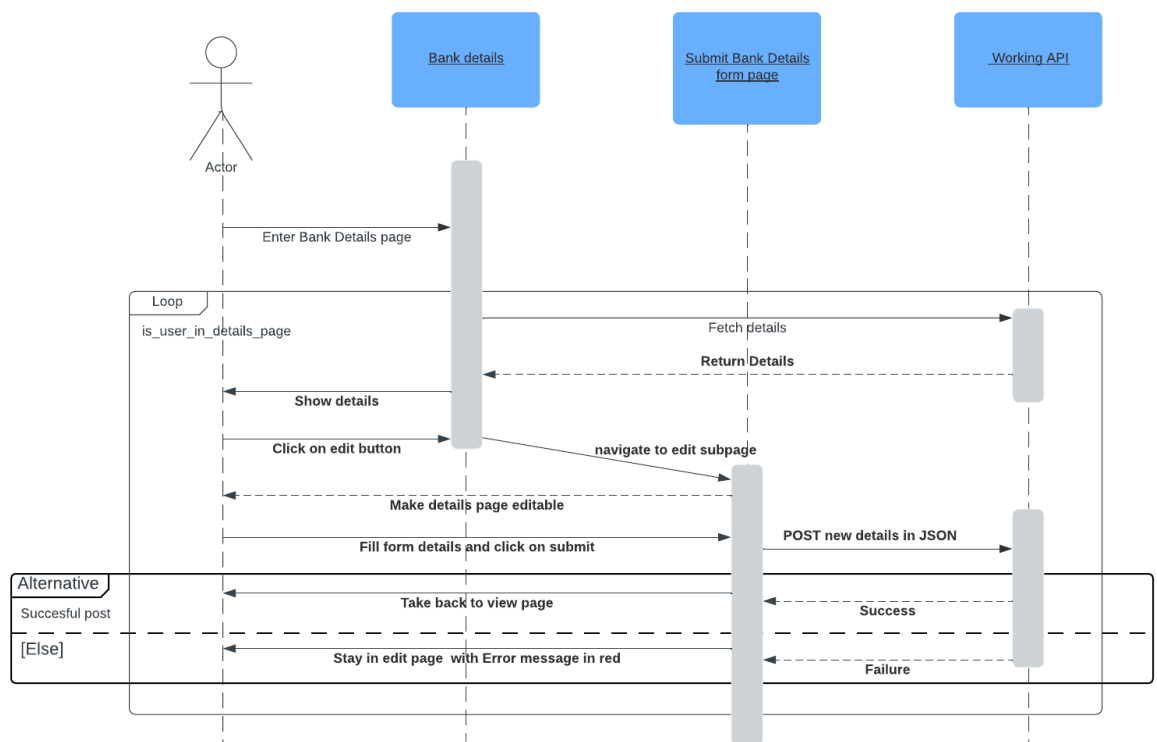
## 1. AUTHENTICATION:

<https://lucid.app/documents/view/f4d8ba08-6da9-4685-ac57-7821d25057d5>



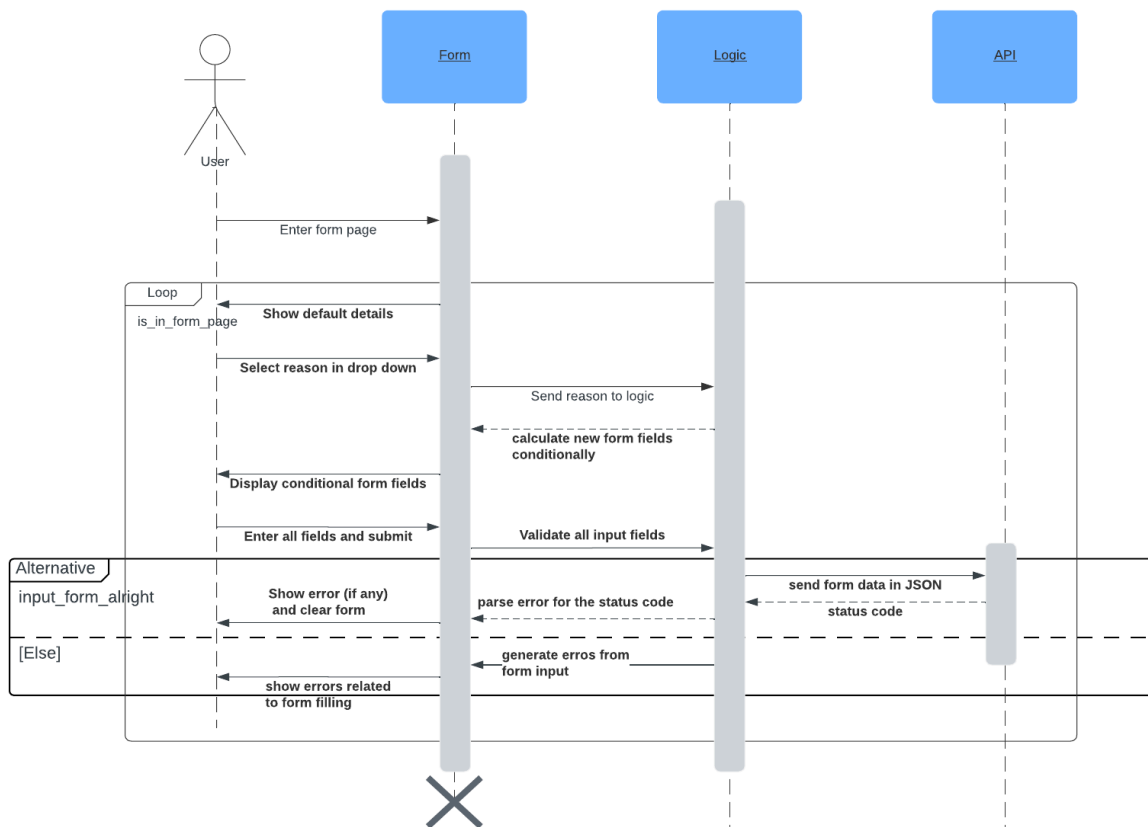
## 2. Fetch details / Post edited details in Bank:

<https://lucid.app/documents/view/194daf9a-f88e-420a-8cb2-c61fa8eb9b06>



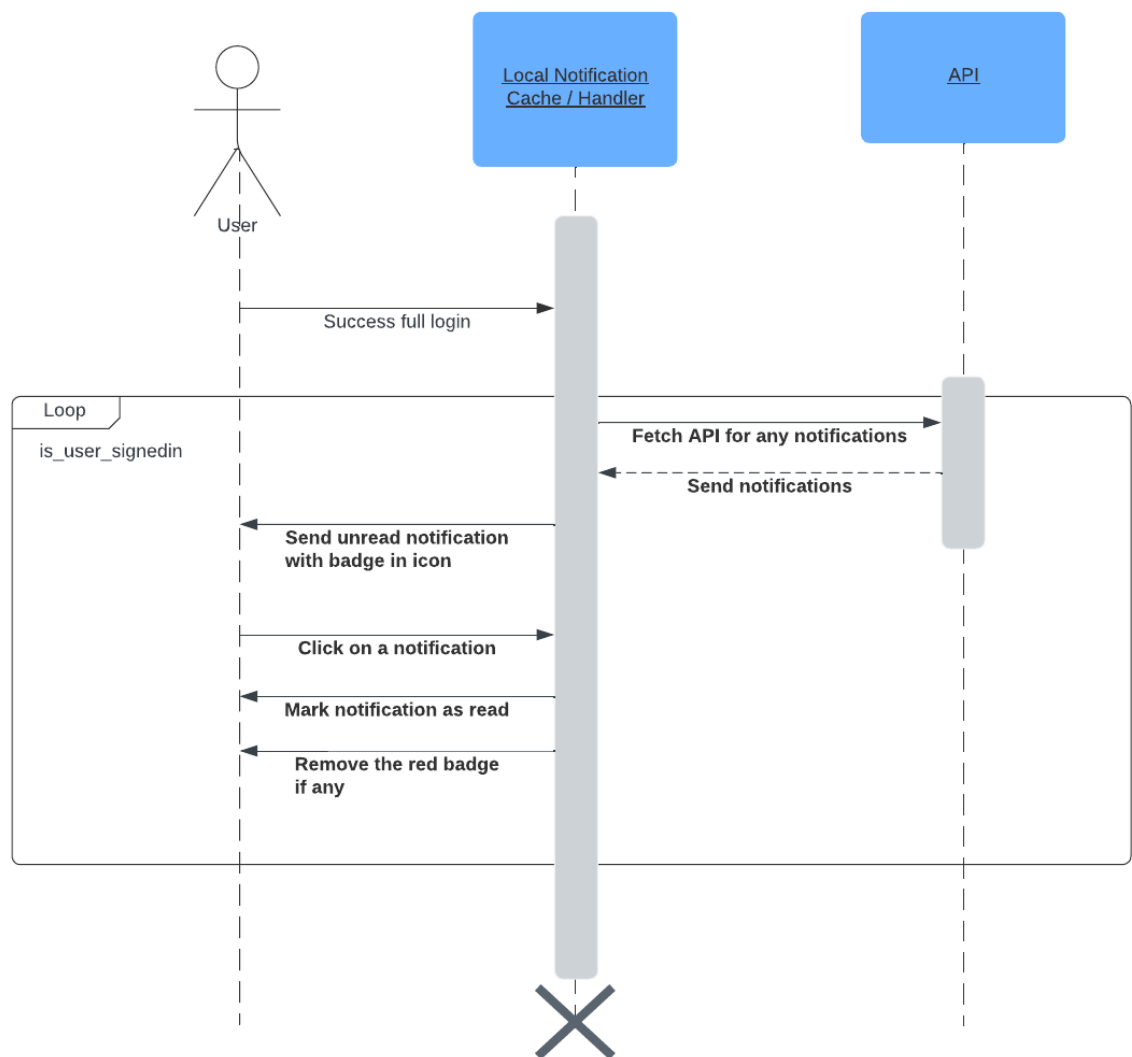
### 3. Leave Application page (form):

<https://lucid.app/documents/view/fa6e4443-460c-4337-8cd8-2745dbf572bd>



#### 4. Notification Handling:

<https://lucid.app/documents/view/c6c17f56-41a9-4c4c-a380-a783b6fb4202>





# Design Rationale

These are the list of issues that may arise during the project are :

1. Users might forget to set up the VPN, risking the security of their data in the app.
2. If a user's internet turns off suddenly after logging in, they might face issues reconnecting.
3. Users may accidentally block file access when attaching documents for leave applications.
4. Users who disable notifications might miss important updates and timely information.
5. For implementing states being shared between different components (if one API fetch were to populate the whole app), we needed "Redux" which is not worth the Hassle.

The Solutions for these problems are:

1. We'll send users a reminder to connect to the VPN when they log in. They won't be able to use the app until the VPN is on, keeping their data safe.
2. We'll show users an error page when they come back, gently letting them know they need to stay online for uninterrupted access.
3. When they try to attach a file, we'll ask them to grant the necessary permission. This ensures a smooth process for submitting leave applications without any hiccups.
4. We'll remind users to turn on notifications, so they don't miss updates about their leave applications. This way, they stay in the loop and don't miss out on crucial information.
5. We avoided that by fetching from the API everytime we needed them separately in separate components.