USER INTERFACE LAB

Subject Code: MCAL25

A Practical Journal Submitted in Fulfillment of the Degree of

MASTER IN COMPUTER APPLICATION

Year 2023-2024

By

Mr. KETAN RAVINDRA DAKI

(Application Id- 78749)

Semester-II

Under the Guidance of

Asst. Prof. Dnyaneshwar Deore

Centre for Distance and Online Education

Vidya Nagari, Kalina, Santacruz East –400098.

University of Mumbai

PCP Center

Satish Pradhan Dnyanasadhana College

Off Eastern Express Highway, Dnyanasadhana, Marg,

Thane, 400604

Center for Distance and Online Education

Vidya Nagari, Kalina, Santacruz East – 400098.

CERTIFICATE

This to certify that, **"Mr. KETAN RAVINDRA DAKI"** appearing **Master's In Computer Application (Semester II) Application ID: 78749** has satisfactory completed the prescribed practical of **MCAL25 - USER INTERFACE LAB** as laid down by the University of Mumbai for the academic year 2023-24.

Teacher In Charge	External Examiner	Coordinator – M.C.A

Date:

Place: -

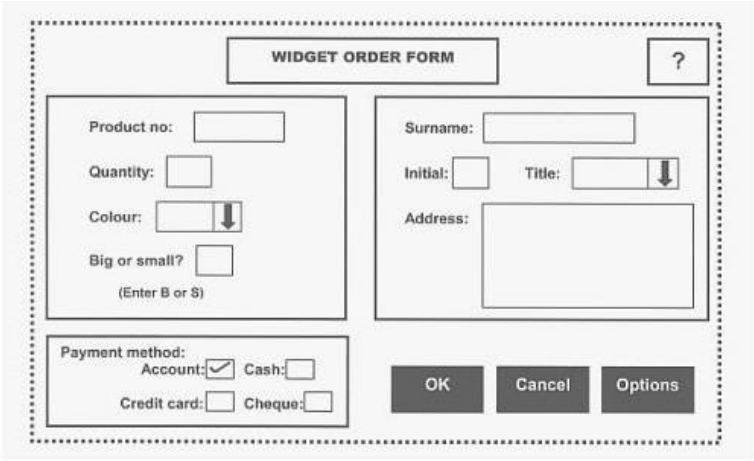
Practical No. 1

Aim:-User interface (UI) design

is likely the first thing you encounter when you use an application or visit a website. User interface design is responsible for a product's appearance, interactivity, usability, behaviour, and overall feel.

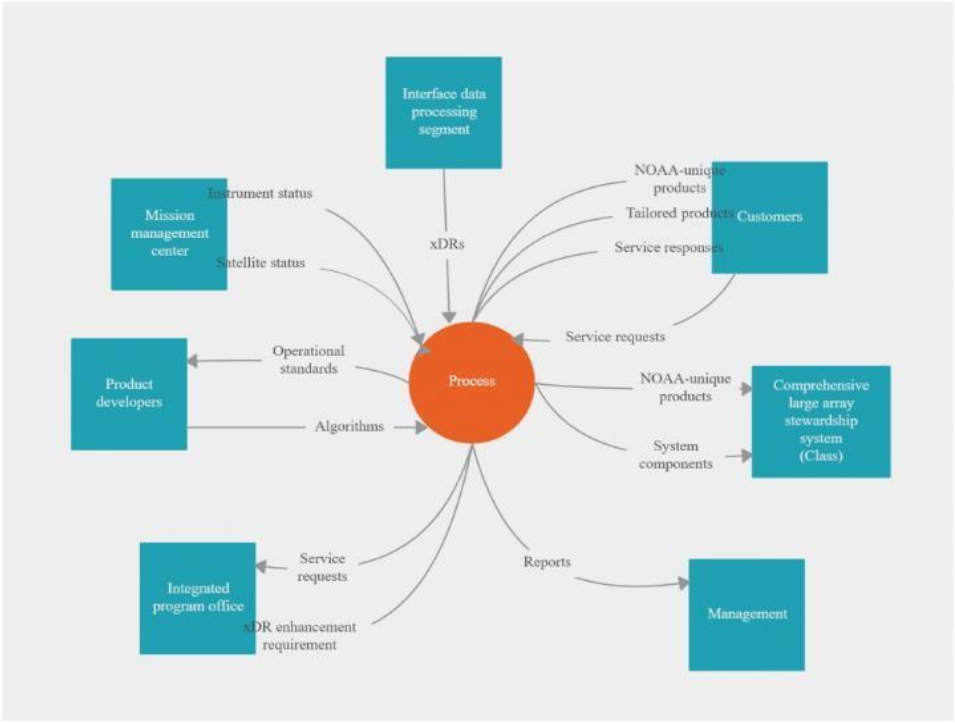
3 main types of UI design	Definition
Graphical user interface (GUI)	A graphical user interface allows users to interact with a device through graphical icons. Typically, these interactions require a mouse, trackpad, or some other point-and-click tool. Your laptop's desktop or home screen is an example of a GUI.
Voice user interface (VUI)	Words and syntax play the most crucial role in voice user interfaces. VUI uses speech recognition to understand voice commands. Notable examples of VUI include iPhone's Siri, Google Home's "hey Google" feature, and Amazon's Alexa.
Menu-driven interface	Menu-driven interfaces provide users with command options via a list or a menu. These commands can present themselves in full-screen or as a pop-up or drop-down. Common examples of menu-driven interfaces include the settings menu on your cell phone. All the user can do is scroll through menus and tap the available items.

Additional types of UI design include touchscreen user interface and form-based user interface. Touchscreen user interfaces are GUIs use touchscreen technology to let you swipe or click rather than needing to use a mouse or stylus. Form-based user interfaces use text boxes, checkboxes, and other informational components. They enable users to fill out electronic forms.


WIDGET ORDER FORM
?
Product no:
Quantity:
Colour:
Big or small?
(Enter B or S)
Surname:
Initial:
Title:
Address:
Payment method:
Account:
Cash:
Credit card:
Cheque:
OK
Cancel
Options

Practical No. 2

Aim:- Requirement gathering flowchart for ui design system



Requirement gathering: Include the business purpose and user needs.

Aim: Project Proposal Digital Wallet

Objective:

● The main objective of the project proposal is to develop a Smartphone application that enables the users to perform any sort of transaction securely through their bank accounts connected to the application over the internet.

● This system is hassle free and more convenient for current modes of transactions which are performed physically.

● Improve wallet apps with more security features like roll-backing a

transaction and reporting a transaction as fraud.

Specifications/Modules:

1. Send/Receive MoneyThis is the main task of this app. Users can send and receive money from

their contacts who're using this application. The system will create a

transaction for each money transfer operation.

User Interface Lab

12

2. Request MoneyUsers can request for money to their contacts and the receiver user will get

a message to send the requested amount or not. If yes then the requested

amount will be deducted from the user.

3. View Past TransactionsUsers can view all the passed transactions they've made along with the

transaction details like transaction id, timestamp, sender, amount etc.

4. Pay Later FunctionIf the user is having insufficient money in his wallet and they're making a

transaction then that transactions will be added to Pay later transactions

history and the equivalent amount will be deducted after a person adds

money in their the wallet

5. Rollback TransactionsIf the user sees some transactions that are not made by them or if they see

any malicious activities with their account (password change emails, otps

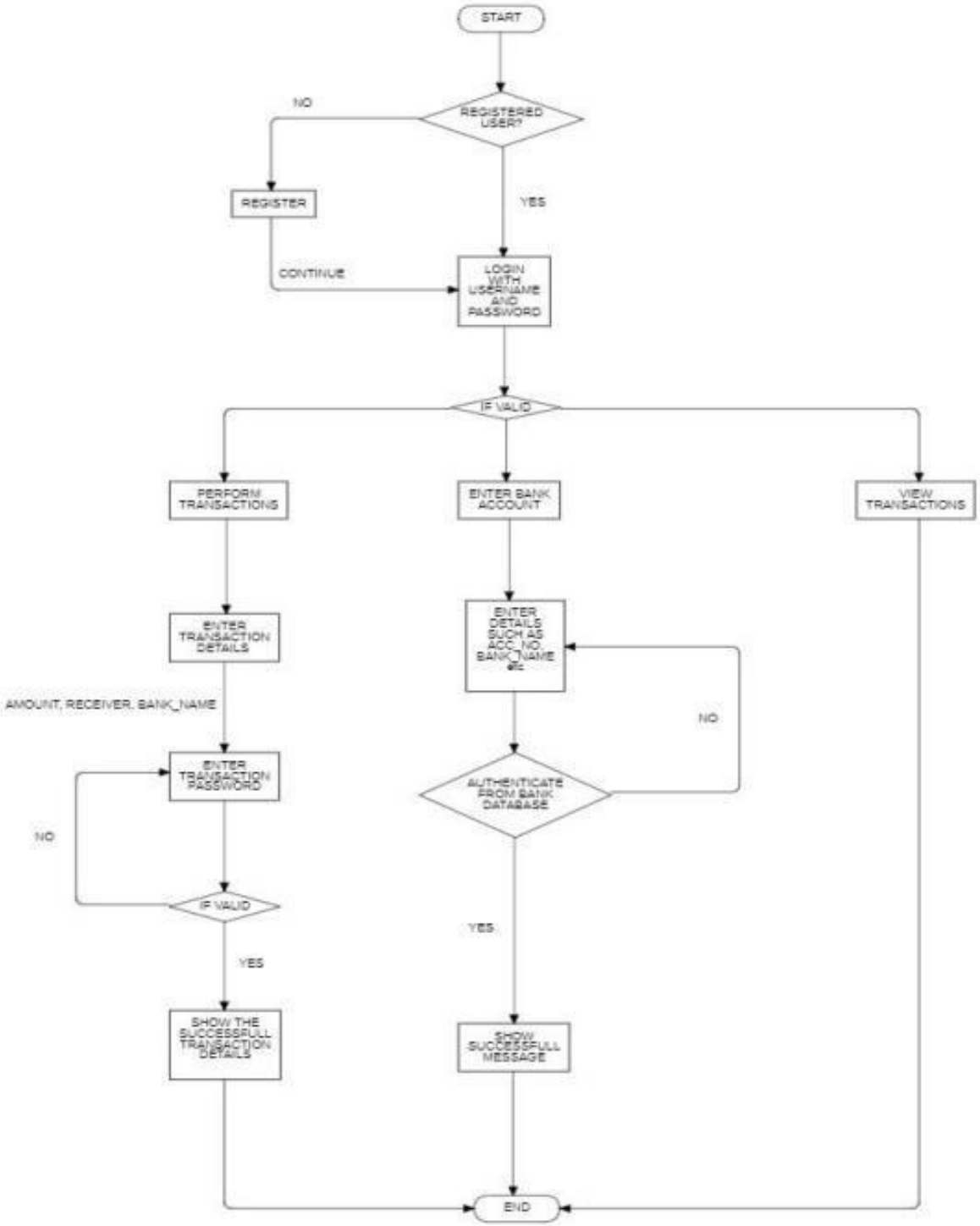
etc) they can request for a rollback transaction in the support.

6. User ProfileUser can update their profile details like email, phone number and change

their password

7. Add/Remove Bank AccountUsers can add and remove new bank accounts from the app.

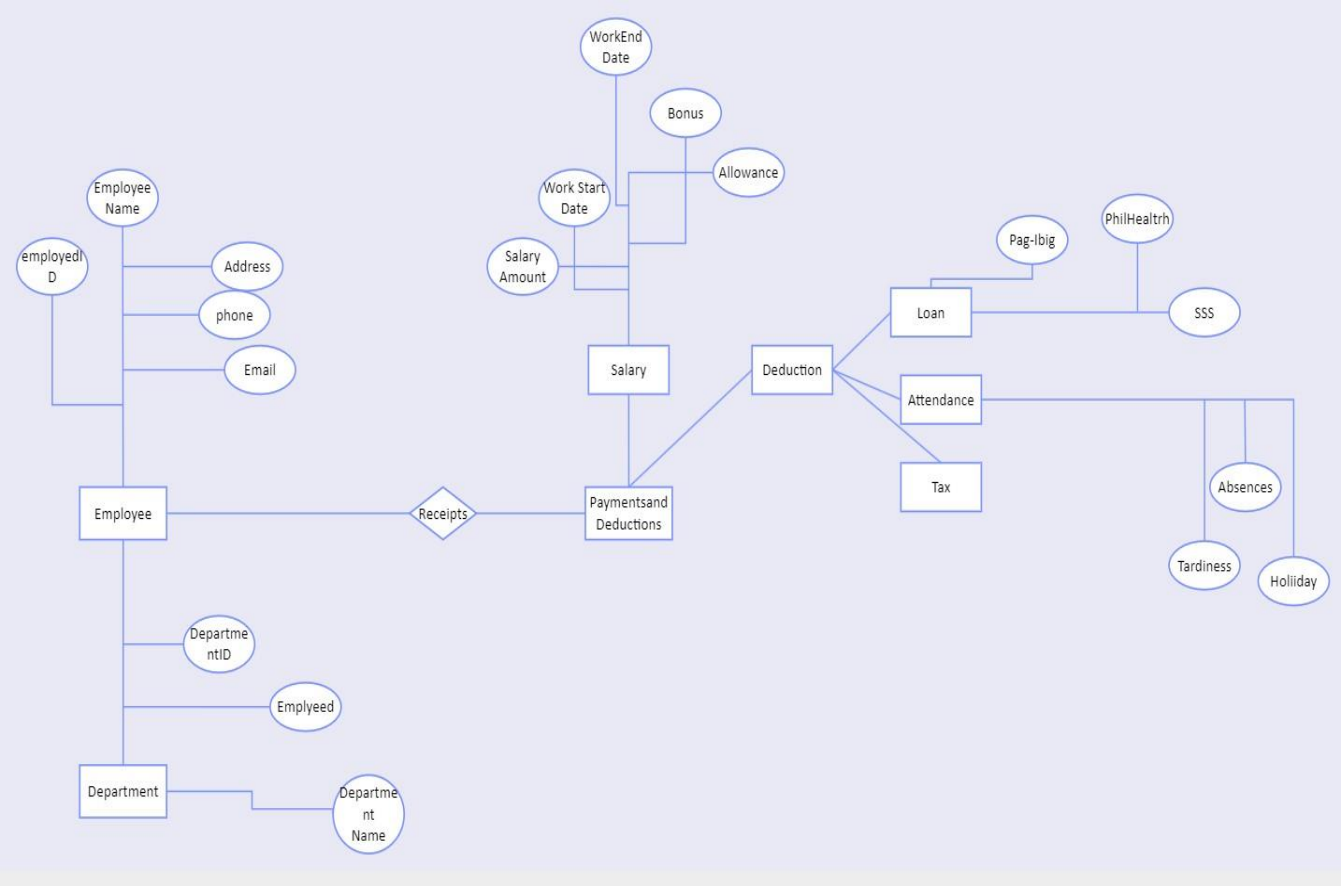
Created using Framer

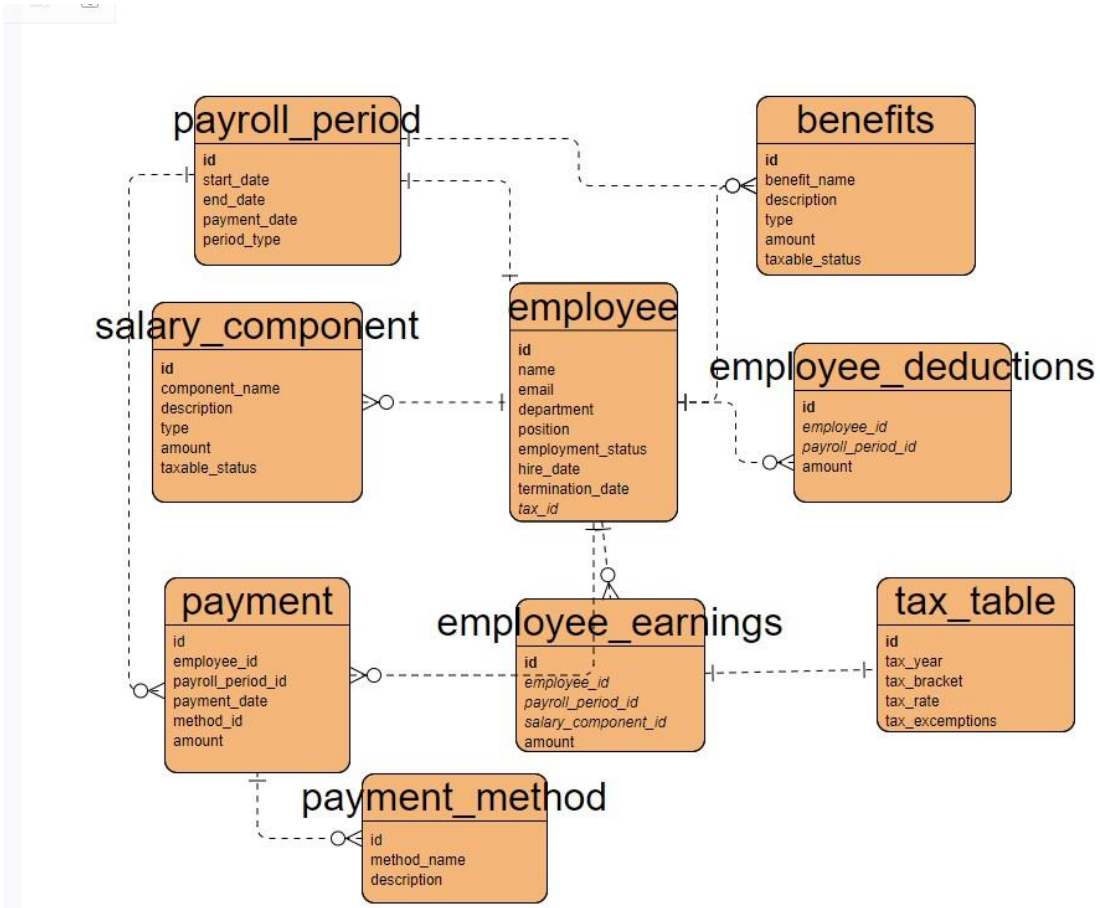

START
REGISTERED USER?
NO
YES
REGISTER
CONTINUE
LOGIN WITH USERNAME AND PASSWORD
IF VALID
PERFORM TRANSACTIONS
ENTER BANK ACCOUNT
VIEW TRANSACTIONS
ENTER TRANSACTION DETAILS
AMOUNT, RECEIVER, BANK_NAME
ENTER TRANSACTION PASSWORD
NO
IF VALID
YES
SHOW THE SUCCESSFULL TRANSACTION DETAILS
ENTER DETAILS SUCH AS ACC_NO BANK_NAME etc
NO
AUTHENTICATE FROM BANK DATABASE
YES
SHOW SUCCESSFULL MESSAGE
END

Practical No. 3

Aim:-Prepare an ERD Diagram for Payroll Management System

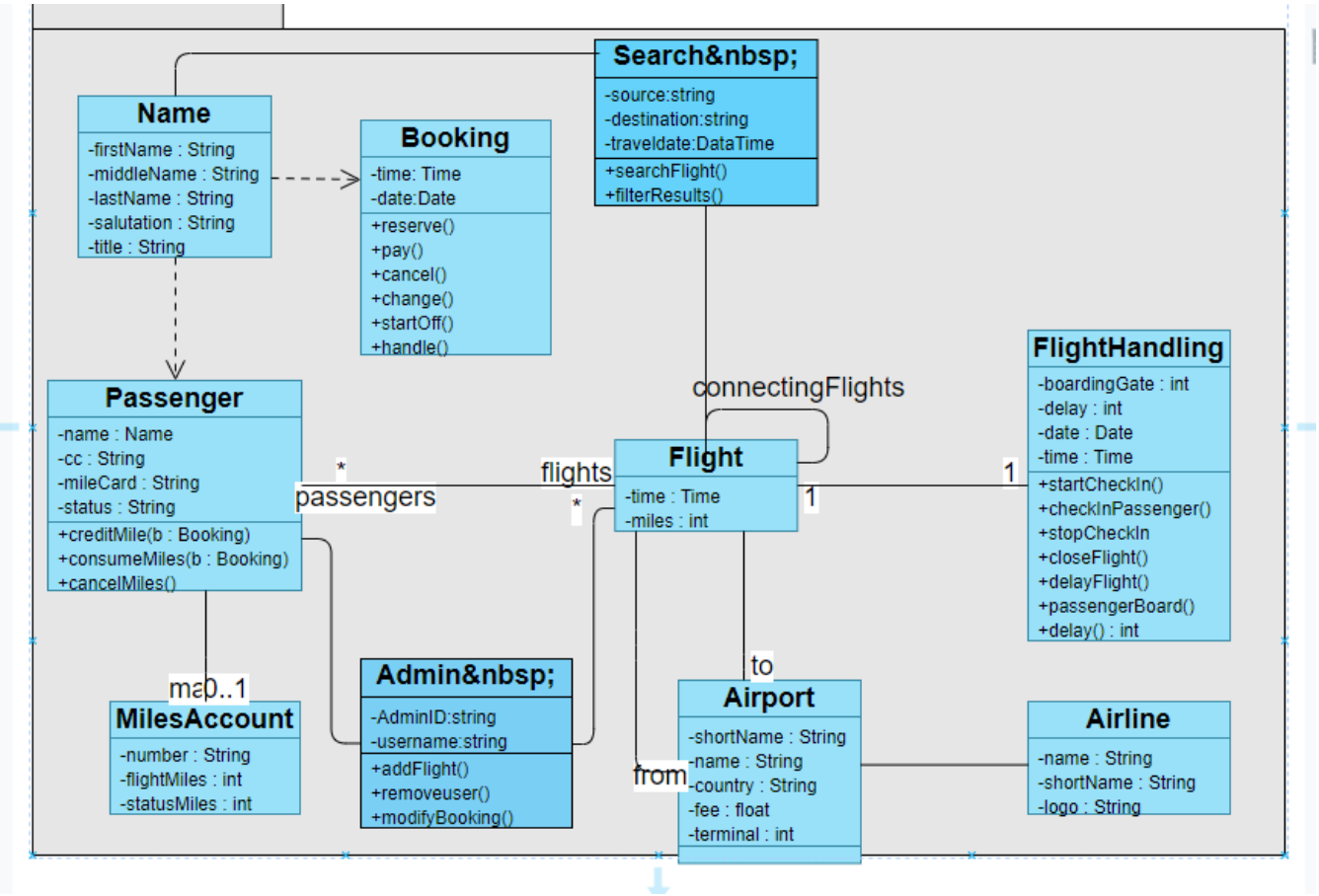
An ER diagram for an employee payroll management system, as the name implies, depicts the relationship between two or more entities. This could be real-world entities or database entity sets. In software development, the model aids in the creation of a logical structure for the database or the overall connectivity.




payroll_period
id
start_date
end_date
payment_date
period_type
benefits
id
benefit_name
description
type
amount
taxable_status
salary_component
id
component_name
description
type
amount
taxable_status
employee
id
name
email
department
position
employment_status
hire_date
termination_date
tax_id
employee_deductions
id
employee_id
payroll_period_id
amount
payment
id
employee_id
payroll_period_id
payment_date
method_id
amount
employee_earnings
id
employee_id
payroll_period_id
salary_component_id
amount
tax_table
id
tax_year
tax_bracket
tax_rate
tax_exemptions
payment_method
id
method_name
description

Practical No. 4

Aim:-Prepare an UML class diagram for Airline Reservation System



Practical No. 5

Aim:- **Scenario 1: Booking a Flight Ticket**

- **User**: Sarah, a frequent traveler, needs to book a flight from New York to London for a business meeting.
- **Objective**: Sarah wants to find a convenient flight with good departure and arrival times.
- **Steps**:
    1. Sarah visits the airline's website or opens the mobile app.
    2. She enters her departure city (New York) and destination city (London) along with her travel dates.
    3. The system presents a list of available flights, showing departure times, arrival times, airlines, and prices.
    4. Sarah selects a flight that fits her schedule and budget.
    5. She enters passenger details and selects her preferred seat.
    6. Sarah proceeds to payment, enters her credit card information, and confirms the booking.
    7. The system issues an e-ticket and sends a confirmation email/SMS to Sarah.

User Story:

Title: As a traveler, I want to easily book a flight ticket online so that I can plan my travel efficiently.

Acceptance Criteria:

- **Scenario 1: Searching for Flights**
    - Given I am on the airline's reservation system,
    - When I enter my departure city, destination city, and travel dates,
    - Then I should see a list of available flights with details including departure times, arrival times, airlines, and prices.
- **Scenario 2: Selecting and Booking a Flight**
    - Given I have chosen a flight from the list,
    - When I select the flight,
    - Then I should be able to enter passenger details and choose my seat.
- **Scenario 3: Payment and Confirmation**

    - Given I have entered passenger details and selected my seat,
    - When I proceed to payment,
    - Then I should be able to securely enter my credit card information and confirm the booking.
- **Scenario 4: Confirmation**
    - Given I have confirmed the booking and payment is successful,
    - Then I should receive an e-ticket and a confirmation email/SMS with my booking details.

These scenarios and the user story outline the key interactions and requirements for a user booking a flight ticket through an airline reservation system. Each scenario focuses on a specific step in the booking process, ensuring clarity and completeness in functionality.

Practical No. 6

Aim:-Draw the storyboard BY HAND with paper and pencil or a drawing tablet and stylist.

To submit the hand drawn paper, you can scan it, take pictures of it, or give it to Amazon to deliver by drone.

Submission:

Submit a single document containing the hand-drawn storyboard image(s).

Use cases

withdraw cash

pay bill

transfer between accounts

list balance

finished

Withdrawal cash

1. The customer choose withdraw cash option on the screen.
2. The system prompts the customer with a menu of accounts to select. The cash will come from the selected account.
3. The customer choose the appropriate account.
4. The system provides the following menu of cash amounts: 20, 40, 60, 80, 100, and other.
5. The customer chooses the amount
6. The ATM check that the withdrawal amount does not exceed the accounts balance or the daily transaction limit. If a limit is exceeded, then the customer is asked to select an new amount, and this step is repeated.
7. The system dispenses the requested amount.

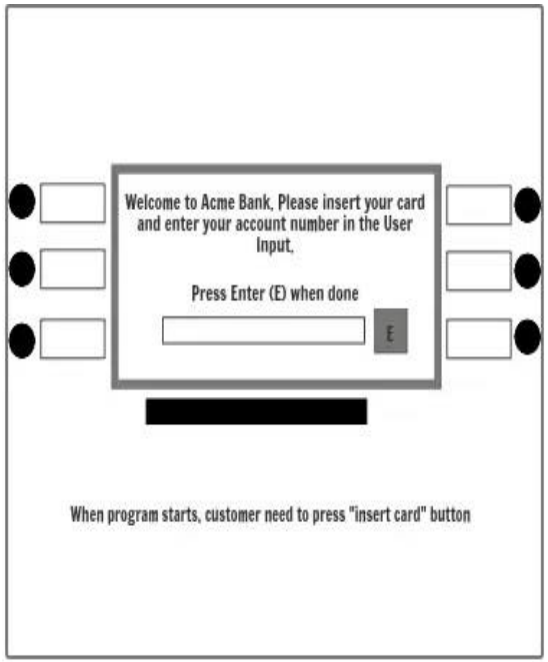
Scenario 6

The customer choose to withdraw 1000$

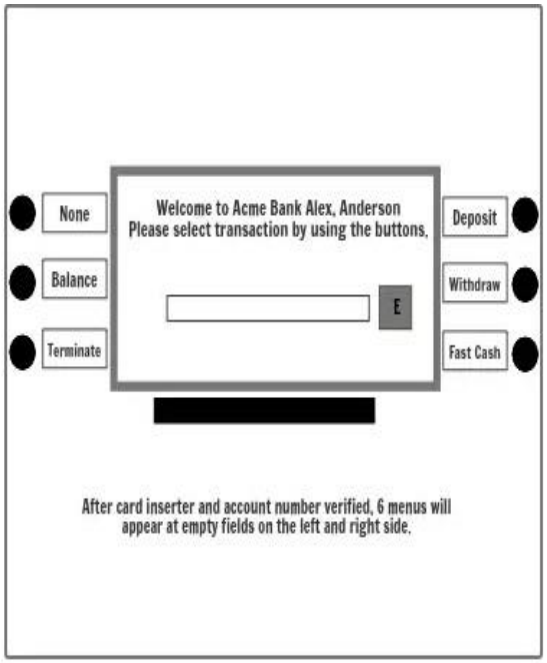
The system reject the requested amount because it exceed the transaction limit

The customer repeat the steps above again

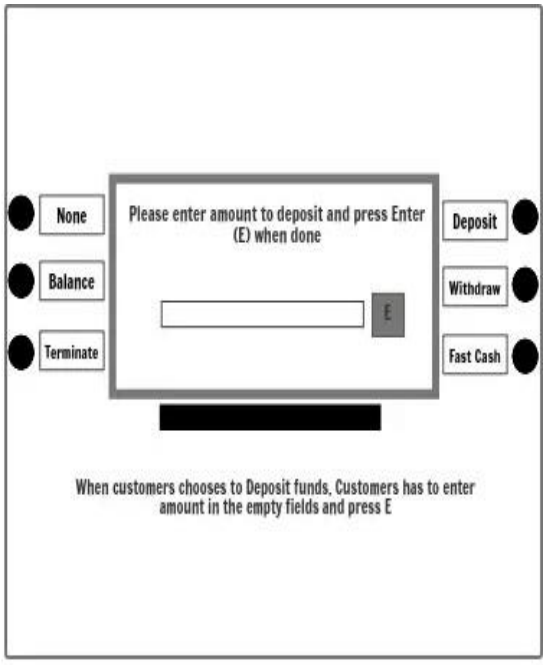
The customer choose and amount less than 1000$



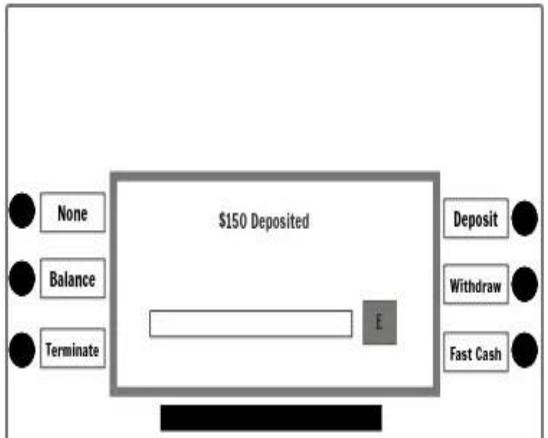
When program starts, customer need to press "insert card" button



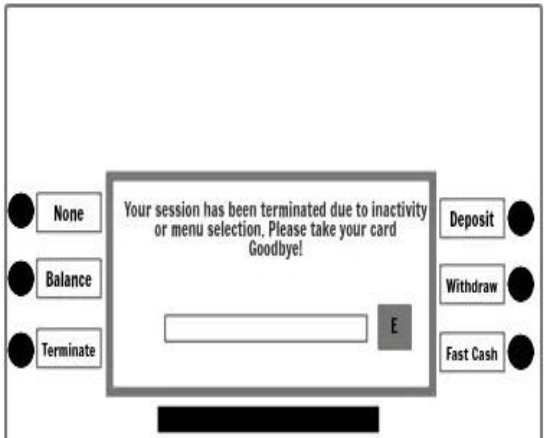
After card inserter and account number verified, 6 menus will appear at empty fields on the left and right side.



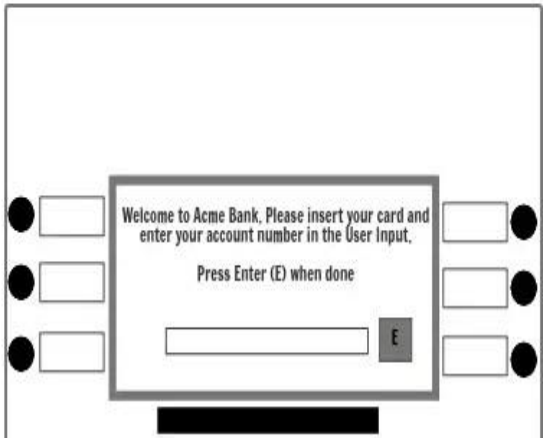
When customers chooses to Deposit funds, Customers has to enter amount in the empty fields and press E



A success message appear with the ammount deposited shown in screen.



When customers chooses "Terminate Button", Program will stops and prompt customer to take their card.



Everything will reset back to Welcome menu.

Practical No. 7

Aim:-Interface/Mockups

A mockup is a static wireframe with much more UI and visual details. If a wireframe is considered as the blueprint of a building, a mockup is similar to a real-life building model. It gives viewers a more realistic impression of how the final website/app will look like, So, it is good for communicating, discussing, collaborating and iterating projects with your team members at a later design stage.

Briefly, unlike wireframes with gray lines, boxes and placeholders, mockups are built with more visual details of the final web/app:

- Rich colors, styles, graphics, and typography
- Actual buttons and texts
- Content layouts and component spacing
- Navigation graphics

WIREFRAME

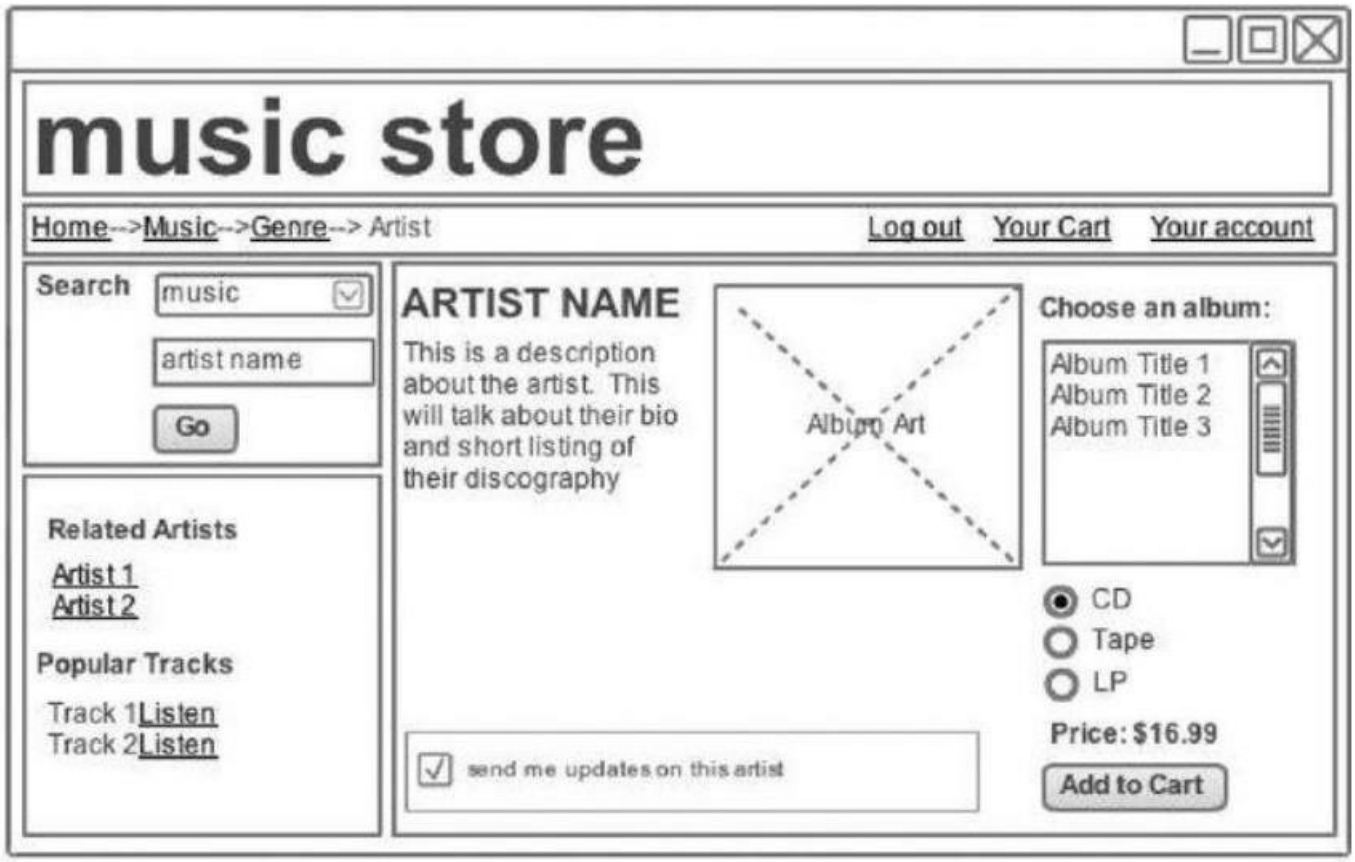
Structure + Functions + Content

MOCKUP

Style + Colours + Right Content

.

Mockup for music store portal



Practical No. 8

Aim:-Functional Calculator in Figma a Prototype Experiment

CHARACTERISTICS/ TRAITS OF USABILITY

UX designers, by their very nature solve problems and seek solutions that creatively align user needs and business goals. A sound solution to a design problem identifies the nature and context of use, whilst taking into account the limitations and constraints in which the resulting product/application will be used.

Based on Jakob Nielsen's 10 Heuristics Principles and typical ATM usage experience, here are some guidelines for ATM designers:

What Designers Should Do:

1. **Visibility of system status**: Ensure users are always informed about the current state of the ATM (e.g., "Processing transaction," "Please wait," "Transaction complete").
2. **Match between system and the real world**: Use familiar concepts and language that users understand easily (e.g., "Withdraw Cash," "Deposit Funds," "Check Balance").
3. **User control and freedom**: Allow users to easily navigate through the options, correct mistakes, and cancel transactions without penalty.
4. **Consistency and standards**: Follow established banking interface standards for layout, navigation, and terminology to reduce cognitive load.
5. **Error prevention**: Design interfaces that minimize the likelihood of errors (e.g., clear button labeling, confirmation screens for critical actions).
6. **Recognition rather than recall**: Present options and instructions clearly on the screen rather than relying on users' memory (e.g., showing available transaction types prominently).
7. **Flexibility and efficiency of use**: Accommodate both novice and expert users by offering shortcuts for frequent tasks (e.g., predefined withdrawal amounts).
8. **Aesthetic and minimalist design**: Keep the interface clean and simple to avoid overwhelming users with unnecessary information or options.
9. **Help users recognize, diagnose, and recover from errors**: Provide clear error messages that explain what went wrong and suggest how to correct it (e.g., "Incorrect PIN entered," "Card not accepted").
10. **Help and documentation**: Include a help option or on-screen guidance for users who may need additional assistance (e.g., accessing account information or using advanced features).

What Designers Should Not Do:

1. **Overcomplicate the interface**: Avoid cluttering the screen with unnecessary options or information that may confuse users.
2. **Use technical jargon**: Minimize the use of banking or technical terminology that may not be familiar to all users.
3. **Require unnecessary steps**: Don't force users to go through multiple screens or steps for common tasks like cash withdrawals.

4. **Hide system status**: Never leave users uncertain about the status of their transactions or the availability of certain functions.
5. **Neglect error handling**: Fail to provide adequate feedback when errors occur or leave users stranded without guidance on how to proceed.
6. **Ignore accessibility**: Overlook the needs of users with disabilities by not providing options for audio instructions, tactile feedback, or clear visual indicators.
7. **Assume user proficiency**: Design with the assumption that all users will understand complex procedures or interfaces without guidance.

Practical No. 9

Aim:-User Interface (UI) and Usability Problems:

Analyze the website www.mu.ac.in, which is presumably the Mumbai University website, and identify potential User Interface (UI) and Usability issues, along with suggestions for improvement. We'll also outline test scenarios and test cases for testing these aspects.

1. **Navigation Issues:**
    - **Problem:** Complex navigation structure, making it hard for users to find specific information.
    - **Suggestions:** Implement a clear hierarchical menu with dropdowns for subcategories. Include a search bar prominently.
2. **Visual Design:**
    - **Problem:** Outdated design, lack of visual appeal, possibly cluttered interface.
    - **Suggestions:** Redesign with a modern, clean interface. Use consistent fonts, colors, and spacing. Ensure accessibility standards are met.
3. **Content Organization:**
    - **Problem:** Information overload on the homepage or sections without clear prioritization.
    - **Suggestions:** Prioritize essential information. Use headings, sections, and summaries to break down content. Consider a "Quick Links" section for popular resources.
4. **Responsive Design:**
    - **Problem:** Lack of responsiveness on mobile devices or smaller screens.
    - **Suggestions:** Implement responsive design principles to ensure the website is usable across all devices. Test thoroughly on various screen sizes.
5. **Accessibility:**
    - **Problem:** Insufficient accessibility features (e.g., alt text for images, keyboard navigation).

- **Suggestions:** Ensure all images have descriptive alt text. Improve keyboard navigation and provide options for text resizing.

6. **Forms and Interactivity:**
   - **Problem:** Complex or confusing forms, lack of interactive elements.
   - **Suggestions:** Simplify forms with clear labels and instructions. Use interactive elements like accordions or tabs for long content.
7. **Page Loading Speed:**
   - **Problem:** Slow loading times due to large images or excessive scripts.
   - **Suggestions:** Optimize images and scripts for faster load times. Use caching and content delivery networks (CDNs) if necessary.

Suggestions for Improvement:

- Conduct usability testing with actual users to gather feedback on navigation, clarity of information, and overall user experience.
- Implement a feedback mechanism for users to report issues or suggest improvements.
- Regularly update content and design based on user feedback and technological advancements.
- Ensure compliance with web standards and accessibility guidelines (e.g., WCAG).

Test Scenarios and Test Cases:

Navigation Testing:

- **Scenario 1:** Verify that main navigation links (Home, About Us, Academics, Admissions, etc.) are visible on every page.
  - **Test Case 1:** Navigate to www.mu.ac.in and verify the presence of main navigation links at the top of the page.
  - **Expected Result:** All main navigation links are visible and functional.
- **Scenario 2:** Test the dropdown functionality of sub-menus under each main navigation item.
  - **Test Case 2:** Hover over the "Academics" menu and verify that sub-menu items (e.g., Departments, Courses) appear.
  - **Expected Result:** Sub-menus expand correctly and display all relevant options.

Responsive Design Testing:

- **Scenario 3:** Ensure the website renders properly on different devices and screen sizes.
    - **Test Case 3:** Access www.mu.ac.in from a mobile device (e.g., iPhone, Android phone) using various browsers (Safari, Chrome).
    - **Expected Result:** Website content adjusts correctly, and all functionalities (navigation, forms) are accessible.

Form Testing:

- **Scenario 4:** Verify the usability and validation of the contact form.
    - **Test Case 4:** Fill out the contact form with valid and invalid data (missing required fields, incorrect email format).
    - **Expected Result:** Form validates input correctly and provides appropriate error messages. Form submission succeeds with valid data.

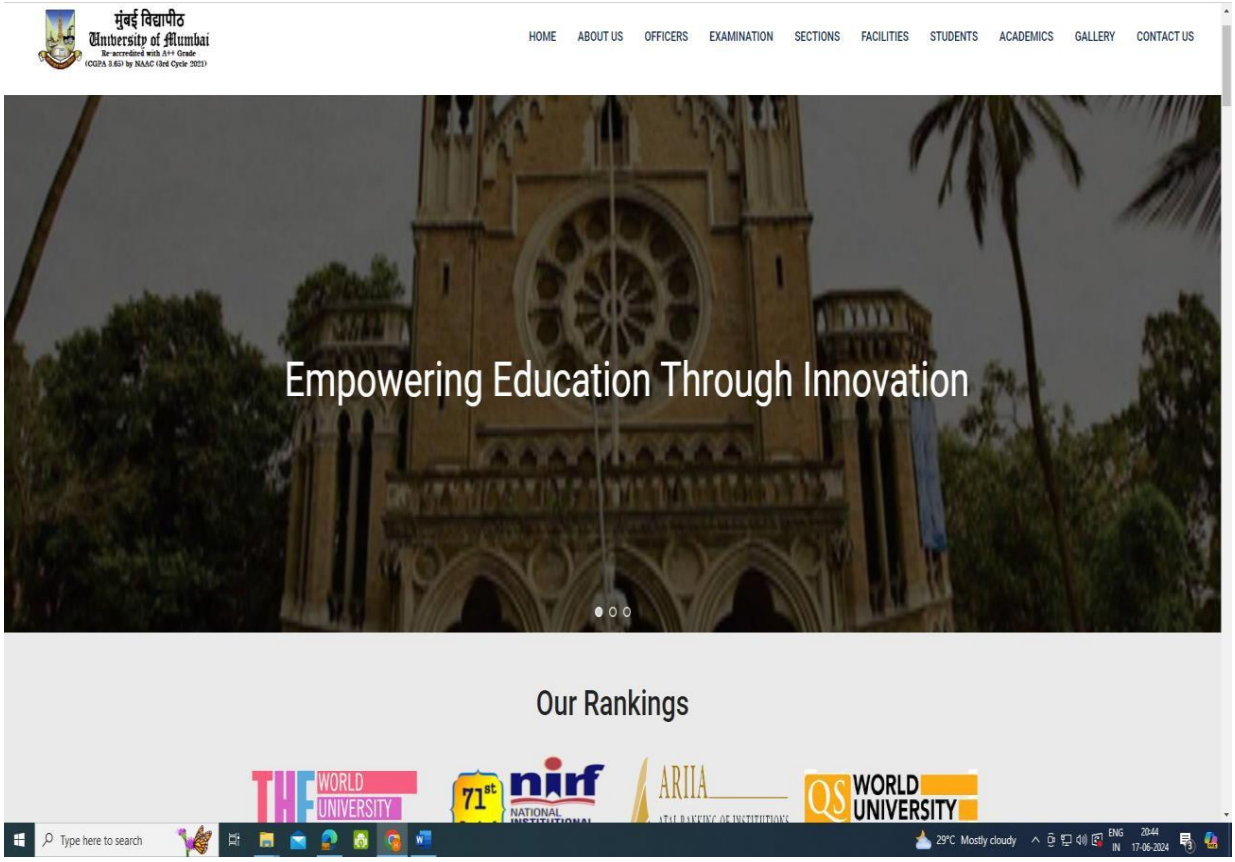
Accessibility Testing:

- **Scenario 5:** Test accessibility features such as keyboard navigation and screen reader compatibility.
    - **Test Case 5:** Navigate through the website using only keyboard keys (Tab, Enter) without using a mouse.
    - **Expected Result:** All interactive elements are accessible via keyboard navigation, and focus is visibly indicated.

Performance Testing:

- **Scenario 6:** Verify the page load speed and performance metrics.
    - **Test Case 6:** Measure the load time of the homepage using tools like Google PageSpeed Insights or GTmetrix.
    - **Expected Result:** Page loads within a reasonable time frame (e.g., under 3 seconds) with optimized content.

These test scenarios and cases cover various aspects of UI, usability, accessibility, and performance, ensuring a comprehensive testing approach for improving the Mumbai University website (www.mu.ac.in).


मुंबई विद्यापीठ
University of Mumbai
Re-accredited with A++ Grade
(CGPA 3.65) by NAAC (3rd Cycle 2021)
HOME
ABOUT US
OFFICERS
EXAMINATION
SECTIONS
FACILITIES
STUDENTS
ACADEMICS
GALLERY
CONTACT US
Empowering Education Through Innovation
Our Rankings
THE WORLD UNIVERSITY
71st
nirf
NATIONAL
ARIIA
QS WORLD UNIVERSITY
Type here to search
29°C Mostly cloudy
ENG
IN
20:44
17-06-2024