**FoodPickup Application**

The "Food in IIIT Hyderabad" application aims to revolutionize the dining experience within the IIIT Hyderabad campus by introducing an efficient, user-friendly mobile application that facilitates digital menu access, pre-ordering, real-time order tracking, secure payment, and feedback submission for the campus canteens and mess facilities.

**Requirements** –

• Digital Menus and Pre-ordering: Enables users to browse and order from up-to-date menus.

• Real-Time Order Tracking: Allows users to view the status of their orders.

• Payment Integration: Supports multiple secure payment methods.

• Feedback Mechanism: Facilitates submission of feedback on meals and services.

**Task 1: Requirements and Subsystems**

Functional and Non-functional Requirements: **<<Akaanksha>>**

Subsystem Overview: **<<Akaanksha>>**

**UX Design – Akanksha**

Stakeholder Identification: **<<Bharti>>**

**Major Design Decisions: <<Bharti>>**

**Architecture** –

Users will interact with the system through a mobile application developed using Flutter, known for its excellent UI capabilities and cross-platform efficiency.

**Design Patterns and Architecture**

• MVC Pattern: For separating the application logic, UI, and data.

• Repository Pattern: To abstract the data layer, making the system more modular and testable. • Singleton Pattern: For database connections and API clients to ensure efficient resource use.

• Observer Pattern: For real-time updates on order status.

**Technical Stack** –

• Frontend: Developed in Flutter for compatibility with both Android and iOS devices.

• Backend: Python with frameworks such as Flask to manage APIs, database interactions, and business logic.

• Database: Use of PostgreSQL or MongoDB, depending on schema requirements.

• Payment Integration: Implementation of secure APIs for payment processing.

**Task 3: Architectural Tactics and Patterns**

**<<NIleema>>**

**Architecture Diagram**

A diagram of a software company

Description automatically generated with medium confidence

**DB model** –

A computer screen shot of a computer

Description automatically generated

**Task 4: Prototype Implementation and Analysis**

**Prototype Development:**

**Work Allocation for Prototype -**

1. **API** 
   1. User Registration – Akanksha
   2. FoodMenu Cart, Order, Menu– Chandana
   3. Feedback, Notification, Address, Payment, Canteen - Nileema
2. **DB Creation – Nileema**
3. **UI – Aditya and Bharti**
4. **Pub-Sub for notification – Aditya**

**Architecture Analysis: <<Aditya>>**

**DB model script** –

@startuml

' hide the spot

' hide circle

' avoid problems with angled crows feet

skinparam linetype ortho

!define primary\_key(x) <b><color:#b8861b><&key></color> x</b>

!define foreign\_key(x) <color:#aaaaaa><&key></color> x

!define column(x) <color:#efefef><&media-record></color> x

!define table(x) entity x << (T, white) >>

table(address) {

primary\_key(addressid) : UUID

column(typeofaddress): character varying(120)

column(address\_desc): character varying(120)

column(createddate): datetime

column(updateddate): datetime

foreign\_key(userid): integer <<FK>>

}

table(canteen) {

primary\_key(canteenid): UUID

column(location): character varying(120)

column(canteenname): character varying(120)

column(canteenowner): character varying(120)

column(canteenstatus): character varying(120)

column(createddate) : datetime

column(updateddate) :datetime

}

table(cart){

primary\_key(cartid): UUID

column(orderid): integer

column(cartstatus): character varying(50)

column(cartprice): integer

column(cartusername): character varying(120)

column(createddate): datetime

column(updateddate): datetime

column(cartuserid): integer

}

table(cartmenuitem) {

primary\_key(cartmenuitemid) : UUID

foreign\_key(cartid) : integer <<FK>>

foreign\_key(menuitemid) : integer <<FK>>

foreign\_key(canteenid): integer <<FK>>

column(menuitemquantity) : integer

column(permenuitemprice) : integer

column(totalmenuitemprice): integer

column(createddate): datetime

column(updateddate): datetime

}

table(feedback) {

primary\_key(feedbackid): UUID

foreign\_key(feedbackuserid) : integer <<FK>>

column(feedbackdesc) : character varying(120)

foreign\_key(orderid) : integer <<FK>>

foreign\_key(menuitemid) : integer <<FK>>

column(feedbackdate) : datetime

column(feedbackstatus) : character varying(50)

column(feedbackaction) : character varying(120)

column(feedbackclosureremarks) :character varying(120)

column(feedbackactionuserid) : character varying(120)

column(createddate) : datetime

column(updateddate) : datetime

column(feedbackusername) : character varying(120)

}

table(menuitem) {

primary\_key(menuitemid) : UUID

foreign\_key(canteenid) : integer <<FK>>

column(menuitemdesc) : character varying(120)

column(Permenuitemprice) : integer

column(menuitemstatus) : character varying(50)

column(menuitemtype) : character varying(50)

column(createddate) : datetime

column(updateddate) : datetime

}

table(notification) {

primary\_key(notificationid): UUID

foreign\_key(userid) : integer <<FK>>

column(notificationdesc) : character varying(120)

column(notificationcreateddate) : datetime

column(notificationtype) : character varying(120)

column(notificationstatus) : character varying(50)

column(createddate) : datetime

column(updateddate) : datetime

}

table(order){

primary\_key(orderid): UUID

foreign\_key(userid) : integer <<FK>>

column(orderdate) : datetime

column(cartid) : integer

column(orderstatus) : character varying(120)

column(paymentid) : integer

column(orderprice) : integer

coumn(createdon) : datetime

coumn(updatedon) : datetime

}

table(users) {

primary\_key(userid) : UUID

column(username) : character varying(80)

column(firstname) : character varying(120)

column(lastname) : character varying(120)

column(mobileno) : bigint

column(email) : character varying(120)

column(aadharid) : character varying(120)

column(userrole) : character varying(120)

column(userstatus) : character varying(120)

column(password) : character varying(100)

column(registered\_on) : timestamp

column(updateddate) : timestamp

column(remarks) : character varying(120)

}

table(payment) {

primary\_key(paymentid) : UUID

foreign\_key(userid) : integer <<FK>>

column(mobileno) : bigint

column(amount) : integer

column(bankaccountnumber) : character varying(120)

column(bankname) : character varying(120)

column(FSCIcode) : character varying(120)

column(pincode) : character varying(100)

column(paymentdate) : timestamp

column(updateddate) : timestamp

column(paymentstatus) : character varying(120)

}

order||..|| users

order ||..|| cart

order ||..|| payment

notification }|..|| users

menuitem||..|| canteen

feedback||..|| menuitem

feedback||..|| order

feedback||..|| users

cart||..||order

cart}|..||users

cartmenuitem }|..||cart

cartmenuitem ||..||menuitem

cartmenuitem }|..||canteen

canteen }|..||users

address}|..||users

@enduml