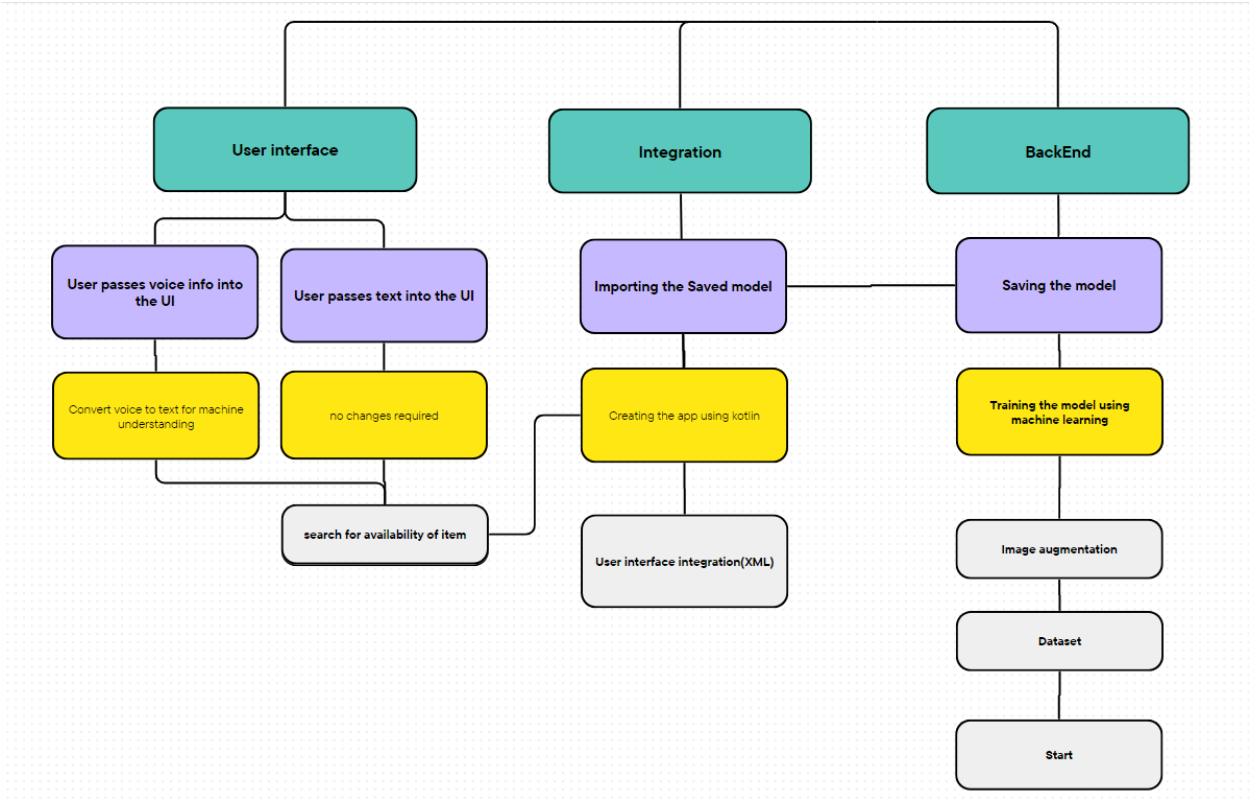
**Project Design Phase-II**
**Technology Stack (Architecture & Stack)**

| | |
|---|---|
| Date | 19/10/23 |
| Team ID | 591057 |
| Project Name | Snack Squad |
| Maximum Marks | 4 Marks |

**Technical Architecture:**




Guidelines:
1. Include all the processes (As an application logic / Technology Block)
2. Provide infrastructural demarcation (Local / Cloud)
3. Indicate external interfaces (third party API's etc.)
4. Indicate Data Storage components / services
5. Indicate interface to machine learning models (if applicable)

Table 1:
Component & Technologies:

| S.No | Component | Description | Technology |
|---|---|---|---|
| 1. | User Interface | How user interacts with | XML layout |

| | | | |
|---|---|---|---|
| | | application e.g.<br>Web UI | |
| 2. | Application Logic-1 | Logic for a process in the application | Kotlin, payment Gateway SDK |
| 3. | Database | Collect the Dataset Based on the Problem Statement | SQLite (Local Storage), Firebase Realtime Database (Cloud Storage) |
| 4. | File Storage/ Data | File storage requirements for Storing the dataset | Local File Storage (e.g., device storage), Cloud Storage (e.g., Firebase Storage) |
| 5. | Frame Work | Used to Create a web Application, Integrating Frontend and Back End | Android SDK (for app development), Google Maps API |
| 6. | Deep Learning Model | Purpose of Model | Optical Character Reading(OCR), Voice Recognition |
| 7. | Infrastructure (Server / Cloud) | Application Deployment on Local System / Cloud<br>Local Server Configuration:<br>Cloud Server Configuration : | Local Server Configuration, Cloud Server Configuration (e.g., AWS, Google Cloud) |

Table 2:
Application Characteristics:

| S.No | Characteristics | Description | Technology |
|---|---|---|---|
| 1. | Open-Source Frameworks | List the open-source frameworks used | Kotlin, Retrofit, Dagger 2 |
| 2. | Security Implementations | List all the security / access controls implemented, use of firewalls etc. | HTTPS, OAuth 2.0, Firebase Authentication |

| | | | |
|---|---|---|---|
| 3. | Scalable Architecture | Justify the scalability of architecture (3 – tier, Micro-services) | Microservices, Load Balancing |
| 4. | Availability | Justify the availability of application (e.g. use of load balancers, distributed servers etc.) | Load Balancers, Distributed Servers |
| 5. | Performance | Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc. | Caching, CDNs, High Request Rate Consideration |