Class Diagram with Data Model

UserData
PK	Username: VARCHAR
	Password: VARCHAR

PR_UserName   FK_UserName

StaffData
PK	StaffID: VARCHAR
	Password: VARCHAR
FK	Username: VARCHAR

Controller
+ run(): void
+ exit(): void

Controller is used to execute the application

UserData and StaffData shows how user information and Staff information are stored and the relation between them.

User
- Username: String
- Password: String
- staffID: String
- UserData: ArrayList<ArrayList<String>>
- StaffData: ArrayList<ArrayList<String>>

+ setUsername(Username: String): void
+ setPassword(Password: String): void
+ Login()
+ StaffLogin()
+ Register(Username: String, Password: String): void
+ RegisterStaff(Username: String, Password: String, StaffID: String): void
+ save(Username: String, Password: String, StaffID: String): void

User logs in to the application first, if they do not have an account, they can register and log in.

On the menu, The user can select submit a feedback. Feedback class receives the feedback and save it into database.

On the menu, the user can select upload the file to the database. Upload class receives the file and save the data into database.

Upload
- fileName: String
- Data: ArrayList<ArrayList<String>>

+ setFileName(fileName: String): void
+ getFileName(): String
+ setData(Data: ArrayList<ArrayList<String>>): void
+ getData(): ArrayList<ArrayList<String>>
+ upload(fileName: String): void
+ analyze(Data: String): void
+ save ():void

Interface
- selection: int

+ MenuDisplay():void
+ setSelection(num: int): void
+ getSelection(): int
+ userSelect(num: int): void

After log in, the menu is displayed

Feedback
- Description: String
- contact: String
- FeedbackData: ArrayList<ArrayList<String>>

+ setDescription(String): void
+ getDescription(): String
+ setContact(String): void
+ getContact(): String
+ feedbacksubmit(Description: String, contact: String): void

Request
- service: String
- Method: String
- address: String
- RequestData: ArrayList<ArrayList<String>>

+setService(Service: String): void
+setMethod(Method: String): void
+setAddress(Method: String): void
+ Request_Submit(service:String, method: String, address: String): void

Search
- keyWord: String
- result: ArrayList<String>

+ setKeyword(String): void
+ getKeyword(): String
+setResult(result: ArrayList<String>): void
+ getResult(): ArrayList<String>
+searchBar(keyWord: String): ArrayList<String>

On the menu, The user can select search bar to search what they want.

API
- APIKey: String
- path: String
- method: String
- parameter: String
- APIRequest: ArrayList<String>
- APIData: ArrayList<ArrayList<String>>

+ setAPIKey(String): void
+ getAPIKey(): String
+ setPath(String): void
+ getPath(): String
+ setMethod(String): void
+ getMethod(): String
+ setParameter(String): void
+ getParameter(): String
+ getAPIRequest(): ArrayList<String>
+ APIReuest(APIKey: String, path: String, method: String, parameter: String): void
+ saveAPI(APIRequest: ArrayList): void

On the menu, the user can select send a request to the database. The request class receives the request and save it into database.

The search class searches data from database and returns it on the screen. Download from the database.

API is used for developer who wants to send API request. API table stores all the API requests

API
PK	APIKey
	Method: VARCHAR
	Path: VARCHAR
	Parameters: VARCHAR

Database
- FeedbackData: ArrayList<ArrayList<String>>
- RequestData: ArrayList<ArrayList<String>>

+ setRequest(request: ArrayList<ArrayList<String>>): void
+ getRequest(): ArrayList<ArrayList<String>>
+ setFeedback(Feedback: ArrayList<ArrayList<String>>): void
+ getFeedback(): ArrayList<ArrayList<String>>
+ save(data: ArrayList<ArrayList<String>>): void
+ DashBoard(data: ArrayList<ArrayList<String>>): void
+ updateDashBoard(data: ArrayList<ArrayList<String>>): void
+ search(keyWord: String): ArrayList<String>
+ download(): void

Database stores data , update data and creates dashboard

Feedback
PK	FeedBackID: VARCHAR
	Description: VARCHAR
	Contact: VARCHAR

Request
PK	RequestID: VARCHAR
	Date: DATE
	Service: VARCHAR
	Address: VARCHAR

Feedback and Request table stores all the feedbacks and the requests