****

**المملكة العربية السعودية**

**وزارة التعليم العالي**

**كليات المعرفة للعلوم و التقنية**

**كلية العلوم التطبيقية**

**قسم علوم الحاسب والمعلومات**

**Kingdom of Saudi Arabia**

**And the Ministry of Higher Education**

**Knowledge Colleges for Science and Technology**

**Faculty of Applied Science**

**Department of Computer and Information Sciences**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**GRADUATION PROJECT**

**Bakkah/ بَكَّة**

**Student names:**

**Abrar Saud ALotaibi 201220173**

**Bushra Ali ALsultan 191220448**

**Shorouq Mubarak ALanzi 191220244**

**Lama khalid alenazi 191220291**

**A report submitted in part fulfilment of the degree of BSc in Information Systems & Health Information Systems.**

**Supervisor:**

**Dr.Nisreen Innab**

**A screenshot of a cell phone

Description automatically generated**

**Department of Computer Science and Information System**

**Riyadh, Kingdom of Saudi Arabia**

[Chapter 1 Problem definition 1-5](#_Toc69331340)

[1.1 Project Description 1-0](#_Toc69331341)

[1.2 Problem Statement 1-0](#_Toc69331342)

[1.3 Objectives 1-0](#_Toc69331343)

[1.4 Scope 1-1](#_Toc69331344)

[1.5 Methodology 1-2](#_Toc69331345)

[1.6 Functional Requirements 1-2](#_Toc69331346)

[1.7 Non-Functional Requirements 1-2](#_Toc69331347)

[1.8 System Context View 1-3](#_Toc69331348)

[1.9 Challenges 1-4](#_Toc69331349)

[1.10 Projections 1-4](#_Toc69331350)

[1.11 Project time line 1-5](#_Toc69331351)

[Chapter 2 System Architecture document 2-7](#_Toc69331352)

[2.1 Introduction 2-0](#_Toc69331353)

[2.2 System Architecture 2-0](#_Toc69331354)

[2.3 Use case diagram 2-0](#_Toc69331355)

[2.4 Class Diagram 2-1](#_Toc69331356)

[2.5 Structural model 2-3](#_Toc69331357)

[2.5.1 Entity Relationship Diagram (ERD) 2-3](#_Toc69331358)

[2.6 Sequence Diagram 2-3](#_Toc69331359)

[2.6.1 User registration sequence diagram 2-4](#_Toc69331360)

[2.6.2 User login sequence diagram 2-4](#_Toc69331361)

[2.6.3 Generating a QR code for Insurance card sequence diagram 2-5](#_Toc69331362)

[Chapter 3 System Design Document 3-6](#_Toc69331363)

[3.1 Introduction: 7](#_Toc69331364)

[3.2 Use case documentation 7](#_Toc69331365)

[3.3 Class diagram documentation 12](#_Toc69331366)

[3.4 Activity Diagram 14](#_Toc69331367)

[3.5 Interfaces 16](#_Toc69331368)

[Chapter 4 Implementation 25](#_Toc69331369)

[4.1 Introduction 26](#_Toc69331370)

[4.1.1 Develop the IT infrastructure \* 26](#_Toc69331371)

[4.1.2 Develop the database and programs \* 26](#_Toc69331372)

[4.2 Implementation approach 26](#_Toc69331373)

[4.2.1 Structured Programming \* 27](#_Toc69331374)

[4.2.2 Object-Oriented Programming \* 27](#_Toc69331375)

[4.3 Development environment 28](#_Toc69331376)

[4.3.1 Android Studio 28](#_Toc69331377)

[4.3.2 Android SDK 28](#_Toc69331378)

[4.3.3 Firebase 29](#_Toc69331379)

[Firebase: The Firebase Real-time Database is a cloud-hosted NoSQL database that lets you store, sync and query data between your users in real-time at global scale. 29](#_Toc69331380)

[4.3.4 Example of Real-time database in Firebase 29](#_Toc69331381)

[4.4 Programming language 29](#_Toc69331382)

[4.5 Mapping from designing to Implementation 30](#_Toc69331383)

[4.6 Code Readability and Maintainability 48](#_Toc69331384)

[Chapter 5 Testing 50](#_Toc69331385)

[5.1 Introduction 51](#_Toc69331386)

[5.2 Mobile application testing issues 51](#_Toc69331387)

[5.2.1 Some of the issues 51](#_Toc69331388)

[5.3 Testing Methodologies 51](#_Toc69331389)

[5.3.1 Functional Testing 52](#_Toc69331390)

[5.3.2 Unit Testing 52](#_Toc69331391)

[5.3.3 Integration Testing 52](#_Toc69331392)

[5.3.4 System Testing 52](#_Toc69331393)

[5.3.5 Acceptance Testing 52](#_Toc69331394)

[5.3.6 Non-Functional Testing 53](#_Toc69331395)

[5.3.7 Performance, Load, Stress Testing 53](#_Toc69331396)

[5.3.8 Security, Vulnerability Testing 53](#_Toc69331397)

[5.3.9 Usability Testing 53](#_Toc69331398)

[5.3.10 Compatibility Testing 53](#_Toc69331399)

[5.3.11 Black Box Testing 53](#_Toc69331400)

[5.3.12 White Box Testing 54](#_Toc69331401)

[5.3.13 Gray Box Testing 54](#_Toc69331402)

[5.4 Test Plan 54](#_Toc69331403)

[5.4.1 Functional Testing 54](#_Toc69331404)

[5.5 Testing tools, Data, Environment 55](#_Toc69331405)

[5.5.1 Testing environment 56](#_Toc69331406)

[5.5.2 Factors for designing Test Environment 56](#_Toc69331407)

[5.5.3 Environmental needs 56](#_Toc69331408)

[5.5.4 Software 56](#_Toc69331409)

[5.5.5 Others 56](#_Toc69331410)

[5.6 Test cases 56](#_Toc69331411)

[5.6.1 User Case 57](#_Toc69331412)

[5.6.2 Security testing 63](#_Toc69331413)

[Conclusions 65](#_Toc69331414)

[Future Work 65](#_Toc69331415)

[References 65](#_Toc69331416)

**Index of figures**

[Figure 1- System Context View 1-4](#_Toc69331423)

[Figure 2-Use case diagram 2-0](#_Toc69331424)

[Figure 3-Class diagram 2-2](#_Toc69331425)

[Figure 4-Entity relationship diagram 2-3](#_Toc69331426)

[Figure 5-Sequence diagram-Register 2-4](#_Toc69331427)

[Figure 6-Sequence diagram-Login 2-4](#_Toc69331428)

[Figure 7-Sequence diagram-QR code generate 2-5](#_Toc69331429)

[Figure 8-Activity Diagram 15](#_Toc69331430)

[Figure 9-Interfaces-FirstPage 16](#_Toc69331431)

[Figure 10-Interfaces-LoginOrReg 17](#_Toc69331432)

[Figure 11-Interfaces-NewReg 18](#_Toc69331433)

[Figure 12-Interfaces-MobileVer 19](#_Toc69331434)

[Figure 13-Interfaces-RegSuccs 20](#_Toc69331435)

[Figure 14-Interfaces-FingerPrintVer 21](#_Toc69331436)

[Figure 15-Interfaces-Login 22](#_Toc69331437)

[Figure 16-Interfaces-MainPage 23](#_Toc69331438)

[Figure 17-Interfaces-QRcodeGenerated 24](#_Toc69331439)

[Figure 18 - Example of Firebase Real-time database 29](#_Toc69331440)

[Figure 19 - Register Sequence Diagram 31](#_Toc69331441)

[Figure 20 - The register operation converted to code 32](#_Toc69331442)

[Figure 21- The register operation converted to code 32](#_Toc69331443)

[Figure 22- The register operation converted to code 33](#_Toc69331444)

[Figure 23- The register operation converted to code 33](#_Toc69331445)

[Figure 24- The register operation converted to code 34](#_Toc69331446)

[Figure 25 - Register interface 35](#_Toc69331447)

[Figure 26 - Login sequence diagram 36](#_Toc69331448)

[Figure 27 - Login operation converted into code 36](#_Toc69331449)

[Figure 28 - Login operation converted into code 37](#_Toc69331450)

[Figure 29- - Login operation converted into code 37](#_Toc69331451)

[Figure 30- Login operation converted into code 38](#_Toc69331452)

[Figure 31- Login operation converted into code 38](#_Toc69331453)

[Figure 32 - Login interface 39](#_Toc69331454)

[Figure 33 - Mobile Verification operation into code 40](#_Toc69331455)

[Figure 34 - Mobile Verification Interface 41](#_Toc69331456)

[Figure 35 - Generating QR code sequence diagram 42](#_Toc69331457)

[Figure 36 - Generate QR code turned into code 42](#_Toc69331458)

[Figure 37- Generate QR code turned into code 43](#_Toc69331459)

[Figure 38- Generate QR code turned into code 44](#_Toc69331460)

[Figure 39- Generate QR code turned into code 45](#_Toc69331461)

[Figure 40 - Generate QR code Interface 46](#_Toc69331462)

[Figure 41- Generate QR code Interface 47](#_Toc69331463)

[Figure 42 - Example of descriptive name 49](#_Toc69331464)

[Figure 43- Testing Methods. 52](#_Toc69331465)

[Figure 44 - User Cases 55](#_Toc69331466)

[Figure 45- User case 57](#_Toc69331467)

[Figure 46 - First Screen 58](#_Toc69331468)

[Figure 47 - Main screen 59](#_Toc69331469)

[Figure 48 - Corrected and Uncorrected Registered. 60](#_Toc69331470)

[Figure 49- Invalid Login 61](#_Toc69331471)

[Figure 50 - Valid Login 62](#_Toc69331472)

[Figure 51- QR code fngerprint passed 63](#_Toc69331473)

[Figure 52 - Security Testing 64](#_Toc69331474)

# Problem definition

## Project Description

## Bakkah to facilitate the transactions of pilgrims during the Hajj season, a wallet is designed to be used for all types of payments, including those made through credit cards. The individual fee can be collected into the Kingdom upon arrival, and the remainder is returned upon departure. Similar to a credit card e-wallet in that it can be linked to a bank account. The electronic wallet mainly contains two components, a program and personal information, and a high security that is not subject to piracy, and a QR is added to facilitate.

## Problem Statement

## The dearth of an ATM near to the Haram, the incapacity to swap different currencies and the availability of support for the aging and those with special protests are a few of the issues that have an impact. Fifth Save time for pilgrims., it decreases the probability of theft Helps reduce the risk of theft.

## Objectives

The main objective of Bakkah is the design of an electronic wallet to preserve the money of pilgrims and provide services that contribute to the goals of the vision in the Kingdom of Saudi Arabia.

The objectives that to be achieved fromBakkah system application are:

* Ease of handling money.
* Easy currency conversion.
* To participate in achieving vision 2030 targets.

## Scope

Analyzing the problem and to assist in drawing, organizing, and knowing the outputs of the system, we need to know the stakeholders.

1. **Clients**
2. **Government (Aljawazat)**
3. **Banks**

The scope will define the boundaries of the application, which include functionalities as the following:

* **User registration**
* To register a new user profile.
* **Technical Linkage with Aljawazat**
* To compare with the Email to verify the identity
* **Email verification**
* To verify the identity of the user
* **QR code generator**
* To generate a QR code every time needed
* **Creating a database to store information**
* **Define the application language**
* **Building the application**
* **Testing for the application**
* **Launching the application**

## Methodology

## The methodology is as follows, the category is determined, and they are Arab expatriates for not knowing the Hajj and Umrah regulations, then the visa number is verified, and then we decide to apply for a special IBAN account number with the bank.

## Functional Requirements

The non-functional requirements are the features to be in the system which used to obtain more efficiency. They include:

**Security**

* + New registration in the app.
  + Connecting the system to a Aljawazat.
  + Connecting the system to a personal email.
  + International bank account.
  + Passwords consist of 8 characters and contain large letters and numbers.
  + The system will carry a user account 7 days when logging in and creating a QR code.
  + Before creating any QR code, there must be a step to verify the user's identity that includes a code sent to email.
  + Every QR code expires after 20 seconds.
  + Every QR code generated is different to enforce security.

**Usability**

* The user should be able to submit any request within 3 or 4 steps maximum.
* The user should be able to learn to use the system within 20 minutes.

**Privacy**

* The app requires the username, password and email sent when login.
* The system must keep information confidential so that other users are not allowed to view the information.

**Availability**

* The system should be available for 24 hours/7 days a week.

**Performance**

* Response time should be fast, so the user should be able to moves between system pages in few seconds**.**
* The system's response time is quick to retrieve and compare the data to generate the QR code in a quick manner.

## System Context View

Person (user) visits a Mecca To perform Hajj or Umrah The application is downloaded through his phone for Create or import e-wallets. The he person (user) logs in to the application. He enters the visa number his to verify the data. Thae's Linking king the IBAN and the money that is added to the wallet is automatically transferred to Saudi riyals. The functions that the user can perform, money can be saved and retrieved in the future and Send money to other addresses.

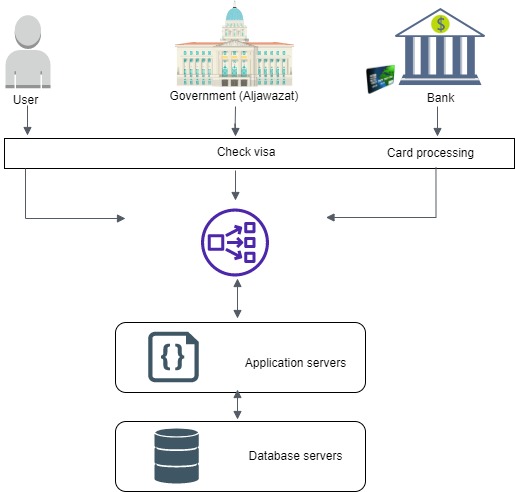


Figure 1- System Context View

## Challenges

There are challenges that we might face in this project like learning how to use some of new methods, skills or programing languages for example:

* Learning a new application coding language to develop the application.
* Looking and using the simplest yet best tools for constructing and analysis to minimize the time cost.
* Communication and linkage with the government (Aljawazat).
* IBAN processing and linkage with the bank.
* Generate QR Code.
* Some research on the project points.

## Projections

## Know the people who are interested in the project idea, and contribute to their own expectations. Implementation of a good appropriate plan, taking care to announce the objectives of the project. Possess a culture that cares about achieving the project's goal. Relying on a committed, efficient and committed team. Building good communication channels. Use case diagram

## From describing the functions of the system from the user's perspective and defining each function of the operator or operator Study of prospective customers The project owner studies the desires and needs of potential customers, their numbers, descriptions, and whereabouts, in order to match the project’s products or services that will be offered in the target market, where knowledge of the customer is the basis for the success of any project Describe the environment in which the system will operate, such as interfaces with power sources, other equipment, software, databases, and users. Safety requirements may be included in the operating environment for the purposes of the system requirements report.

## Project time line

We made a plan for the entire project, including all the sub task that we need to complete in this project, using projectPlan360 tool as shown in the table below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Task | Duration | Start | | Finish | Resource |
| 1 | Bakkah Application | 2 days | Tuesday  6/9/2022 | | Wednesday  7/9/2022 | Bushra , Shorouq  Abrar , Lama |
| 2 | project proposal | 3days | Thursday  8/9/2022 | | Saturday  10/9/2022 | Bushra , Shorouq  Abrar , Lama |
| 3 | **Chapter 1** | **20 days** | **Sunday**  **11/9/2022** | | **Friday**  **30/9/2022** | **Bushra , Shorouq**  **Abrar , Lama** |
| 4 | Project Description | 1 day | Sunday  11/9/2022 | | Sunday  12/9/2022 | Bushra , Shorouq  Abrar , Lama |
| 5 | Problem Statement | 1 day | Monday  12/9/2022 | | Monday  13/9/2022 | Bushra , Shorouq  Abrar , Lama |
| 6 | Objectives | 1 day | Tuesday  13/9/2022 | | Tuesday  14/9/2022 | Bushra , Shorouq  Abrar , Lama |
| 7 | Scope | 1 day | Wednesday  14/9/2022 | | Wednesday  15/9/2022 | Bushra , Shorouq  Abrar , Lama |
| 8 | Methodology | 1 day | Thursday  15/9/2022 | | Thursday  16/9/2022 | Bushra , Shorouq  Abrar , Lama |
| 9 | Functional Requirements | 2day | Friday  16/9/2022 | | Saturday  18/9/2022 | Shorouq |
| 10 | Non-Functional Requirements | 1 days | Monday  18/9/2022 | | Monday  19/9/2022 | Bushra |
| 11 | System Context View | 4 days | Monday  18/9/2022 | | Thursday  22/9/2022 | Abrar |
| 12 | Challenges | 4 days | Monday  18/9/2022 | | Thursday  22/9/2022 | Abrar |
| 13 | Projections | 2 days | Friday  23/9/2022 | | Saturday  24/9/2022 | Lama |
| 14 | Project time line | 1day | | Saturday  24/9/2022 | Saturday  24/9/2022 | Bushra, Shorouq |
| 15 | Presentation | 4 days | | Sunday  25/9/2022 | Friday  30/9/2022 | Bushra , Shorouq  Abrar , Lama |

# System Architecture document

## Introduction

This chapter contains many of important topics with related subjects, these subjects are: (System Architecture, Design decisions, Domain model, Architectural Style, Structural model). To their importance for considered the building blocks for application to be implemented, and finally we will depict some diagrams, to illustrate the functionality of the application.

## System Architecture

## System architecture is a compact, manageable description of how a system is organized and how the components inter operate. The system architecture is often the same for systems with similar requirements.

## Use case diagram

A use case is a method that used in the system analysis to model the requirements of the system. The use case is involving of a group of processes that interact with the systems. The predictable users are show in determined environment and related to a particular objective. It offers a group of components that can be used together. The use case should describe all system activities that have relations to the users as show in Figure**.**

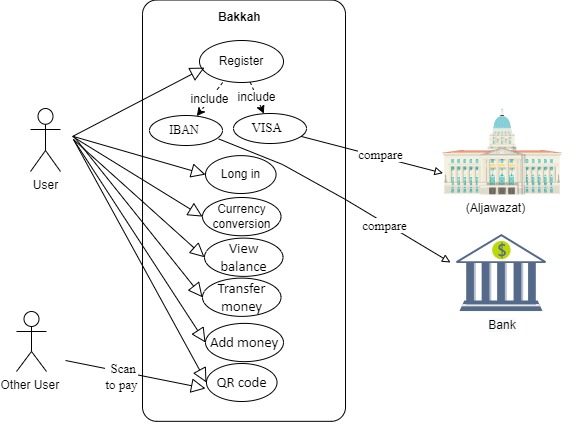
****

Figure 2-Use case diagram

## Class Diagram

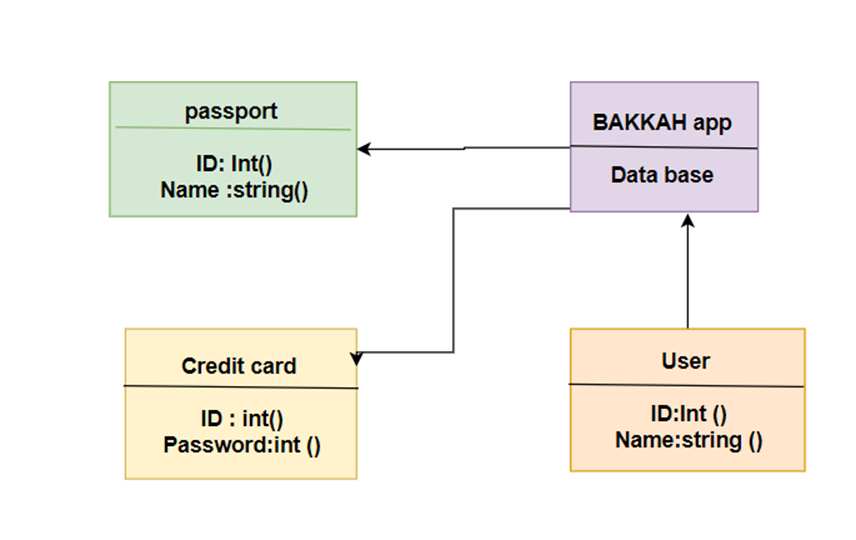


Figure 3-Class diagram

## Structural model

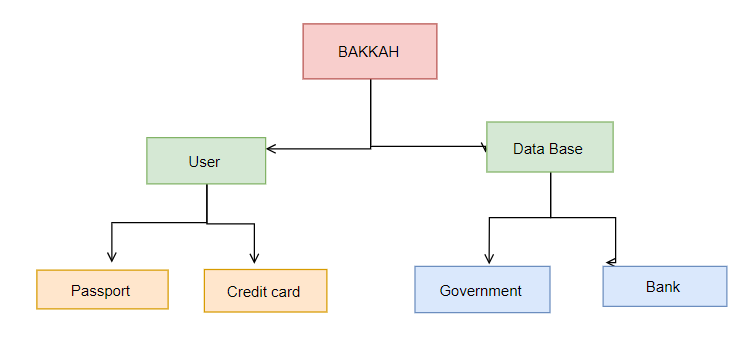
****

Figure 4-structural model

### Entity Relationship Diagram (ERD)

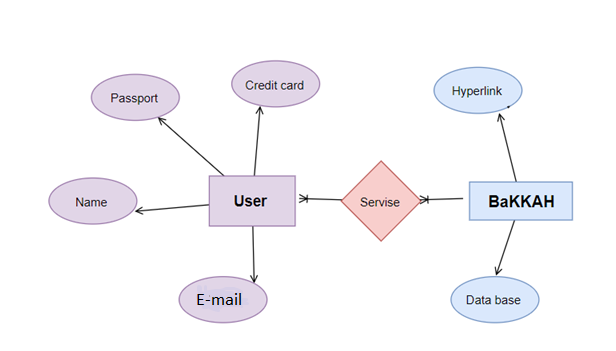
****

Figure 5-Entity relationship diagram

## Sequence Diagram

Sequence diagrams are interaction diagram that displays how processes work with one another and in what order. It is a construct of a Message Sequence. A sequence diagram displays object interactions arranged in time sequence. It shows the objects and classes involved in the scenario and the sequence of messages exchanged between the objects. It needed to carry out the functionality of the scenario as show in Figures

### User registration sequence diagram

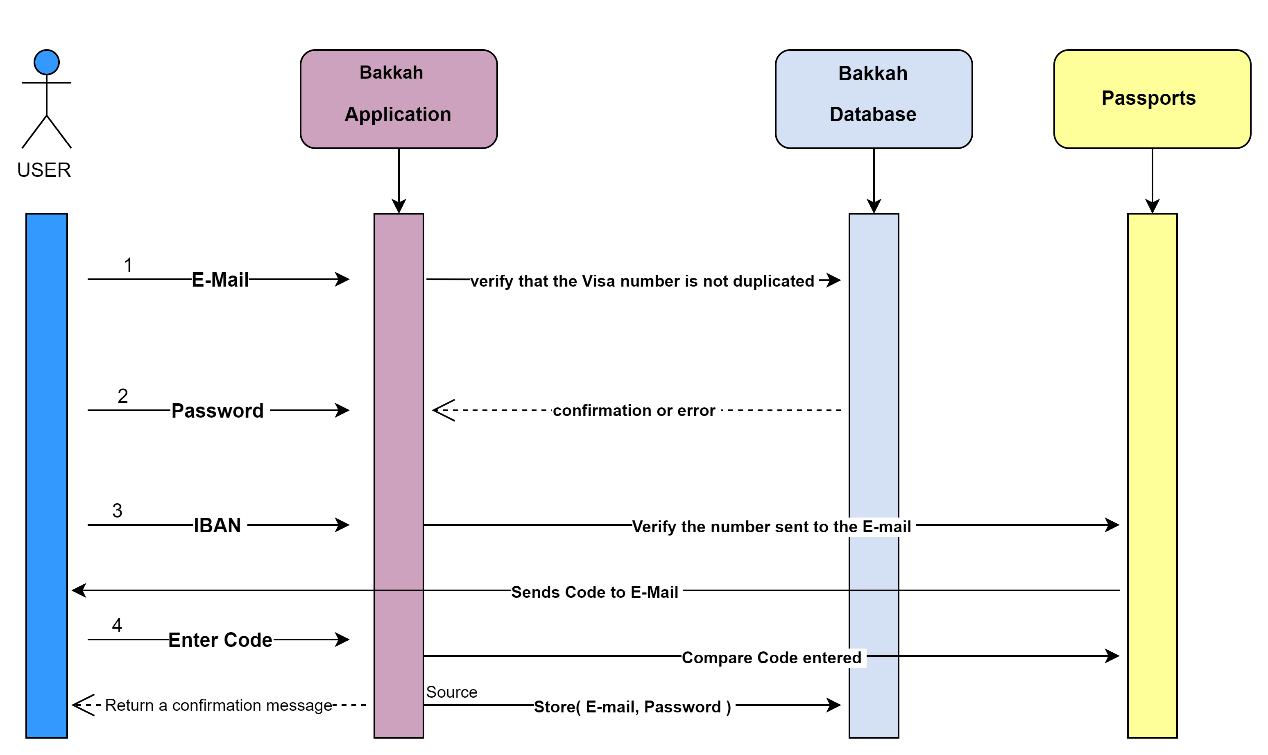


Figure6 -Sequence diagram-Register

### User login sequence diagram

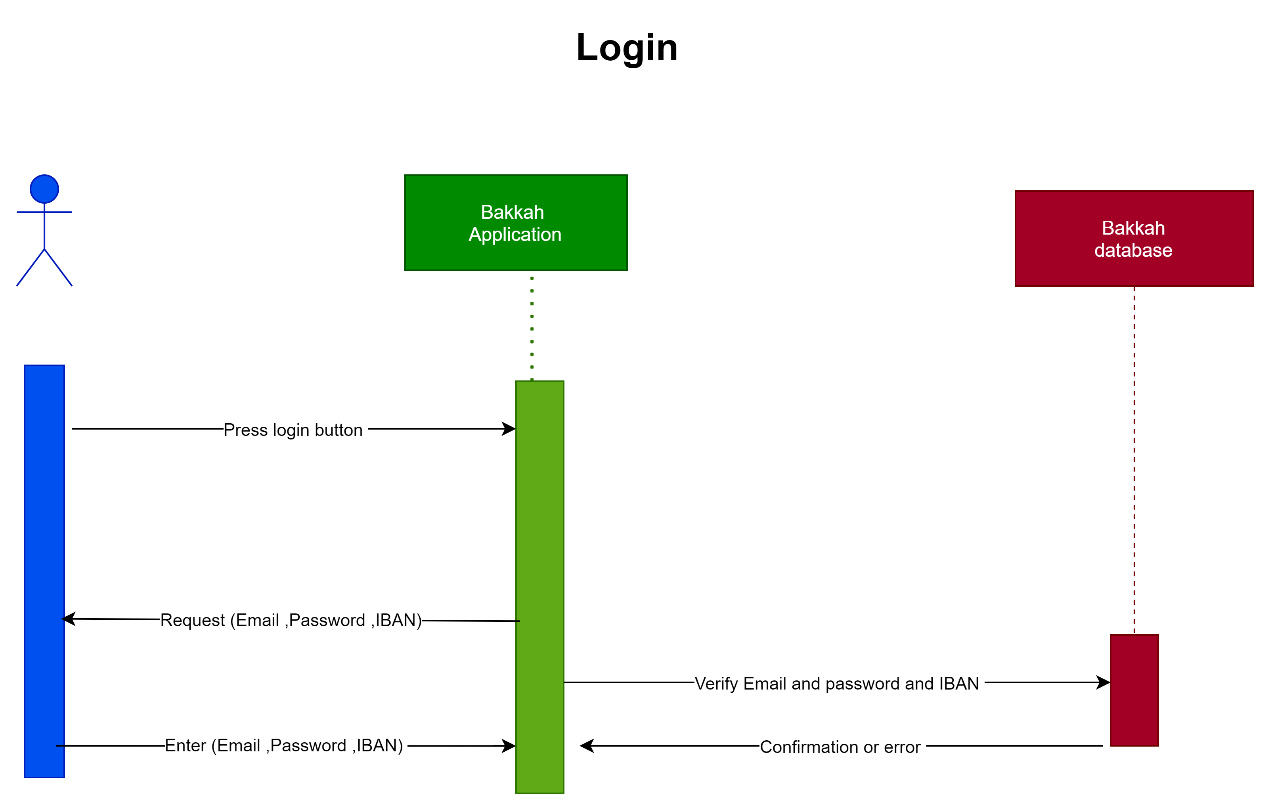


Figure 7-Sequence diagram-Login

### Generating a QR code for Insurance card sequence diagram

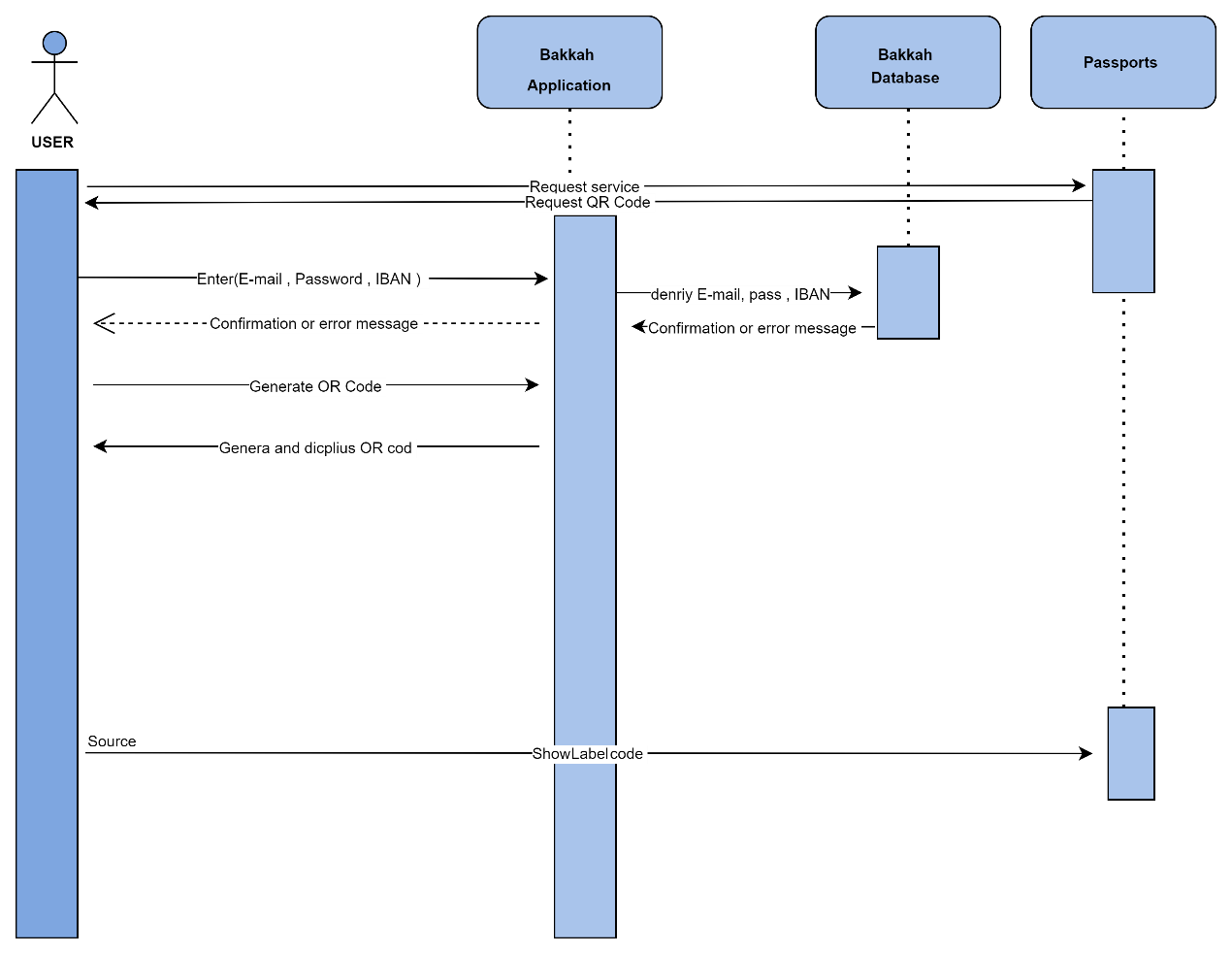


Figure 8-Sequence diagram-QR code generate

# System Design Document

## Introduction:

## This section focuses on completing the system's illustrations and explains the use case description as well as the system interfaces for each user and also contains implementation consideration.

## Use case documentation

|  |
| --- |
| Register |

|  |  |
| --- | --- |
| Actors | User |
| Objective | Creating a new User account in the database |
| Pre-condition | The filled information should be checked and validated. |
| Post-condition | The new user account after the condition is stored in the emaildatabase. |
| Primaryscenario | User presses Register button.User enters all the required information.User presses Submit button.The system checks the filled information.The system requests the code sent to the user from the Passports.The system responses with a confirmation message and creates the account. |
| Alternativescenario | User presses Register button.User enters all the required information.User presses Submit button.The system checks the filled information.The system asks for the code sent to the user from passports.The system responses with an error message about some filled information. |

|  |
| --- |
| link email |

|  |  |
| --- | --- |
| Actors | User |
| Objective | Link the created account to your personal email |
| Pre-condition | Prerequisite Email must be verified |
| Post-condition | The email is linked to the user's account |
| Primaryscenario | The system asks the user to put a personal email to link it to his account.User sets personal email.The system verifies the personal email.The system responds with a confirmation message and links the personal email to the account. |
| Alternativescenario | 1. The system asks the user to put a personal email to link it to his account. 2. User sets personal email. 3. The system verifies the personal email.  The system responds with a confirmation message and links the personal email to the account. |

|  |
| --- |
| Login |

## 

|  |  |
| --- | --- |
| Actors | User |
| Objective | Login to the system |
| Pre-condition | The filled information should be checked and validated |
| Post-condition | The user enters the system |
| Primary scenario | The user fills in the personal email and password2-The system checks the filled in information3- The user logs in successfully |
| Alternative scenario | 1. The user fills in the personal email and password2. The system checks the filled in information3. The system shows an error message and asks theuser to re-enter his information |

|  |
| --- |
| Generating QR code |

|  |  |
| --- | --- |
| Actors | User |
| Objective | Generate a QR code |
| Pre-condition | IBAN verification step for personal bank account |
| Post-condition | The user generates a QR code |
| Primary scenario | 1. The user logs into the system successfully2. User clicks Generate QR Code3-The system checks whether the email entered matches the email saved in the system previously4-The system retrieves the information and generates a QR code |
| Alternative scenario | 1. The user logs into the system successfully2. User clicks Generate QR Code3-The system checks whether the email enteredmatches the email saved in the system previously4-The system retrieves the information and generates aQR codeThe system returns an error message |

## Class diagram documentation

**User class:**

Attributes:

* Visa number: The visa number of the person.
* Password: The password of the account.

Methods:

* Verify login: This method does the login operation to the application
* Logout: This method does the logout operation from the application
* Compare Code: This method compares between the mobile code sent to the user with the code that the user entered.

|  |  |
| --- | --- |
| Class | User |
| Attributes | * National ID: The national ID of the person * Password: The password of the account |
| Methods | * Verify login: This method does the login operation to the application * Logout: This method does the logout operation from the application * Compare Code: This method compares between the mobile code sent to the user with the code that the user entered |

**QR class:**

Methods:

* Generate: This method generates the QR code
* Display: This method displays the QR code
* Terminate: This method terminates the QR code

|  |  |
| --- | --- |
| Class | QR |
| Methods | * Generate: This method generates the QR code * Display: This method displays the QR code * Terminate: This method terminates the QR code |

**Class Insurance:**

Methods:

* Retrieve Data: This method retrieves the needed data to generate the QR code from the insurance company

|  |  |
| --- | --- |
| Class | Insurance |
| Methods | •Retrieve Data: This method retrieves the needed data to generate the QR code from the insurance company |

**Class Aljawazat:**

Methods:

* Verify mobile number: this method sends a code to the user's mobile number

|  |  |
| --- | --- |
| Class | ABSHER |
| Methods | * Verify mobilenumber: this method sends a code to the user's mobile number |

## Activity Diagram

The activity diagram is the way to show what are the basic different operations of the system, In addition to describing the behavior of the system during these different operations. It explains how to follow the operations from beginning to end in an orderly way step by step.

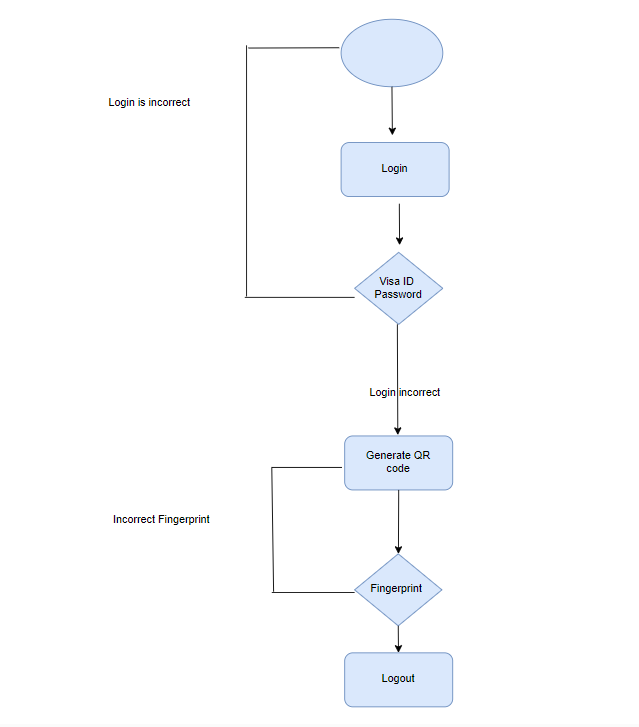


Figure9-Activity Diagram

## Interfaces



Figure10-Interfaces-FirstPage

This interface shows the first page that appears when the user opens the application.

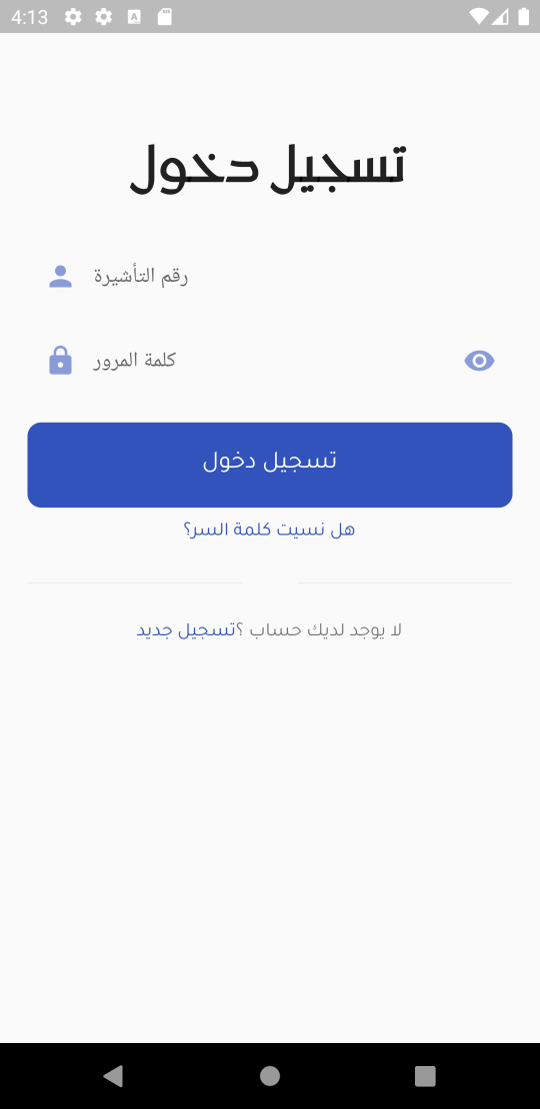


Figure 11-Interfaces-LoginOrReg

This interface shows two buttons, the first one is to login and the second one is to register.

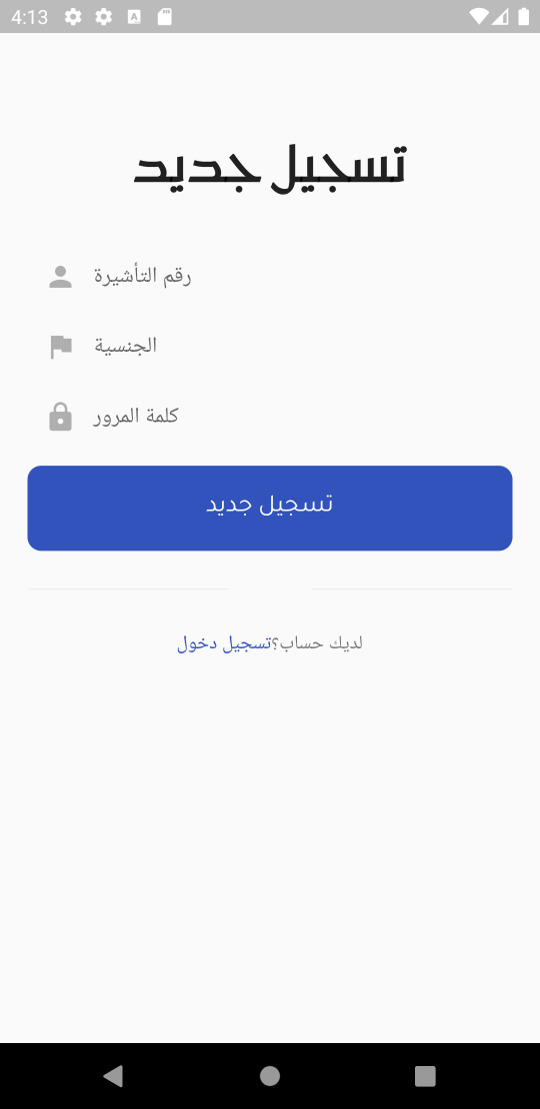


Figure 12-Interfaces-NewReg

Shows a Register screen. In this figure, this interface dedicated to registration in application, Registration is done by filling the following fields: Visa ID, Nationality, password.

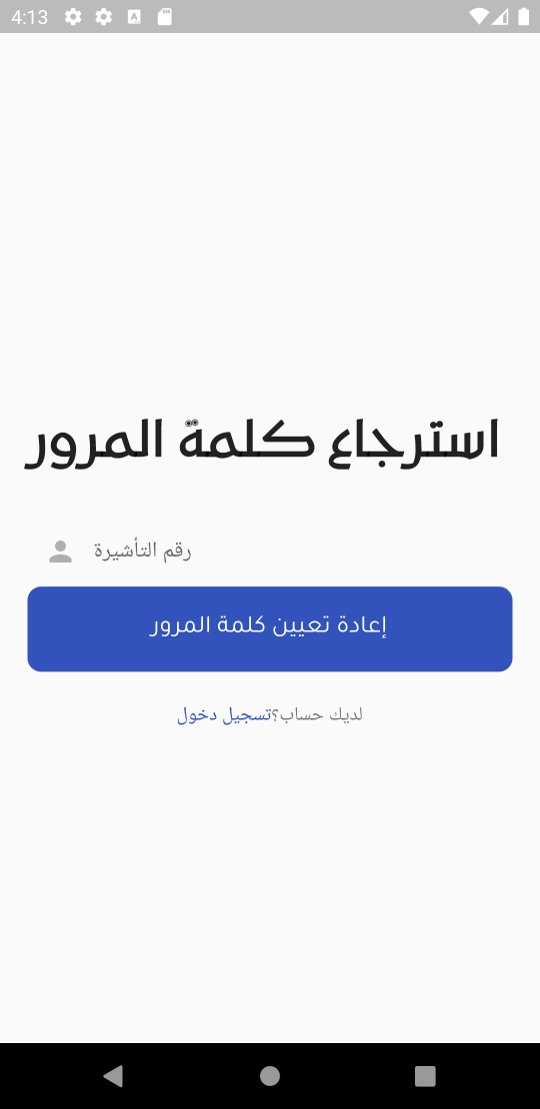


Figure13 -Interfaces-MobileVer

This interface shows the Email verification step for the registration.



Figure 14-Interfaces-RegSuccs

This interface shows that the registration has been completed successfully.

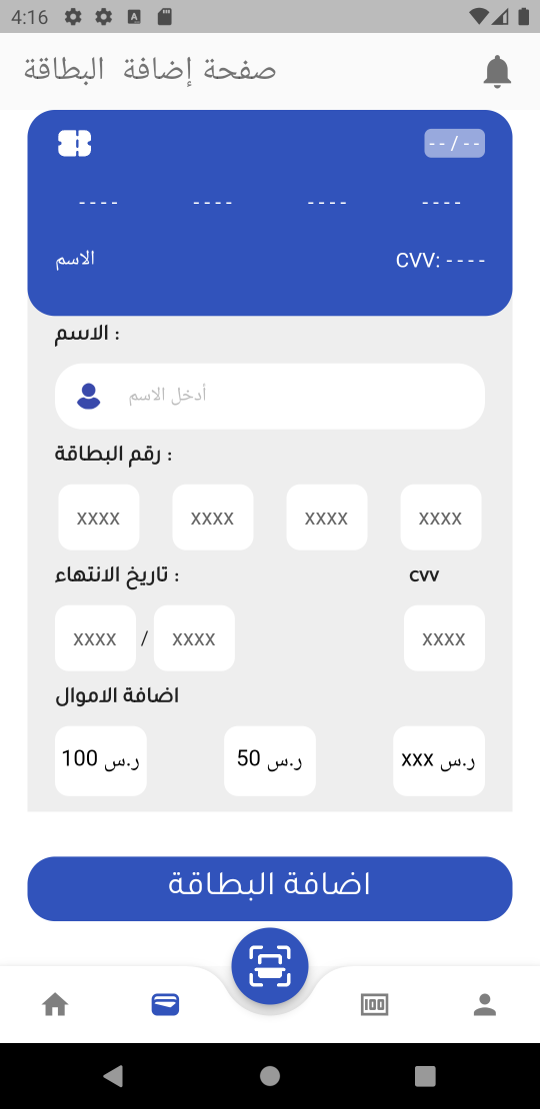


Figure 14-Interfaces- Bank Card

This interface shows the bank card entry and name, the application will ask to verify after registration to link it to the registered account and after logging in.

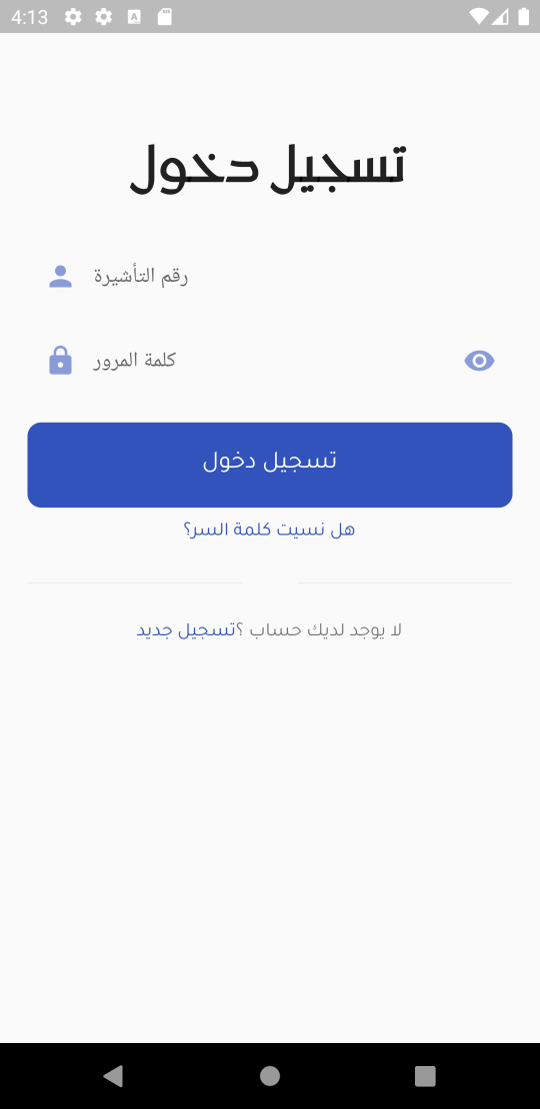
s

Figure16 -Interfaces-Login

Shows a Login screen. In this figure, this interface dedicated to log on to the application, you are logged in by entering your Visa ID and password in the allotted places followed up with a Email.



Figure 17-Interfaces-MainPage

This interface shows the main page that contains Generate QR code



Figure 18-Interfaces- Billing page

This interface shows the personal funds that were used during the stay period.

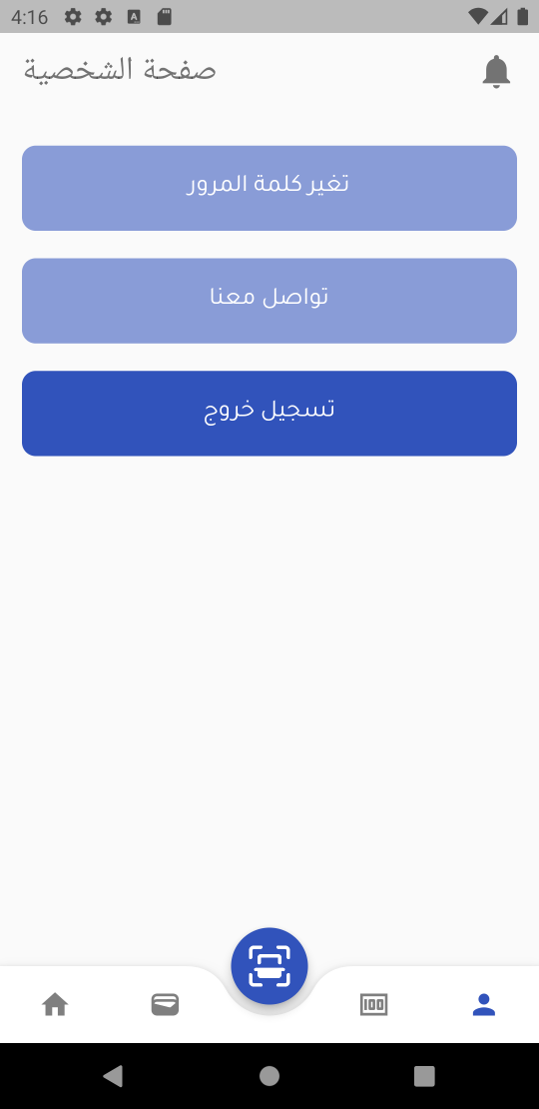


Figure19-Interfaces- Billing page

This interface shows personal information, money.

# Implementation

## Introduction

The fourth phase in the development of any system or project is the implementation phase after the accomplishment of the planning, analysis and design phases according to the System Development Life Cycle (SDLC). The implementation phase includes taking all of the detailed design documents from the design phase and converting them into the real system.

The two primary actions involved in the implementation phase are:

1. Develop the IT infrastructure.
2. Develop the database and programs.

### Develop the IT infrastructure \*

The platform upon which the system will work must be constructed before building the actual system. In the design phase, an organization makes a proposal of the planned IT infrastructure displaying the design of the software, hardware, and telecommunication equipment. In the implementation phase, the organization obtaining and implements the essential equipment to support the IT infrastructure. Most new systems require new hardware and software. It may be as simple as adding memory to a client or as complex as setting up a comprehensive area network across several states.

### Develop the database and programs \*

When the IT infrastructure is made, the organization can begin to create the database and write the programs essential for the system. IT specialists do these functions and it can take months or even years to design and generate all the needed elements to complete the system.

## Implementation approach

In our project, we will use two types of programming are: structured and object oriented programing.

### Structured Programming \*

Structured programming is a subgroup of procedural programming that imposes a logical structure on the program being written to type it more effective and easier to comprehend and alter. Structured programming often employs a top-down design model, in which developers map out the overall program structure into distinct subsets. A defined function or set of similar functions is coded in a different module or sub module, which means that code can be loaded into [memory](http://searchmobilecomputing.techtarget.com/definition/memory) more powerfully and that modules can be reused in other programs. After a module has been verified individually, it is then combined with other modules into the whole program structure.

### Object-Oriented Programming \*

Object-oriented programming is a programming language model prepared

A[bout objects](http://searchsoa.techtarget.com/definition/object) rather than actions and data rather than logic. In history, a program has been seen as a logical procedure that takes input data, processes it, and yields output data. The programming challenge was seen as how to write the logic, not how to describe the data. Object-oriented programming takes the vision that what we really care about are the objects we want to handle rather than the logic required to manipulate them. Examples of objects range from human beings (defined by name, address, and so forth) to constructions and floors (whose properties can be defined and managed) down to the little [widgets](http://whatis.techtarget.com/definition/widget) on a computer desktop (such as buttons and scroll bars). Object-oriented programming does not emphasis on structure but emphases on data modeling. Examples of object-oriented programming languages include C, Java, HTML and PHP.

## Development environment

## We will use Flutter is a mobile app development platform created by Google. It allows us to create cross-platform apps that run on Android and iOS devices. Flutter uses a reactive programming language called Dart, making development faster and easier than traditional methods.

## We download and install some software to start coding such as:

### GIT

Git is a DevOps tool used for source code management. It is a free and open-source version control system used to handle small to very large projects efficiently.

### Visual Studio Code

Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging.

### Android Studio

The Device Manager is a tool can launch from Android Studio that helps create and manage AVDs. The Device Manager is a tool that defines the characteristics of an Android phone that want to simulate in the Android Emulator.

### Example of Login

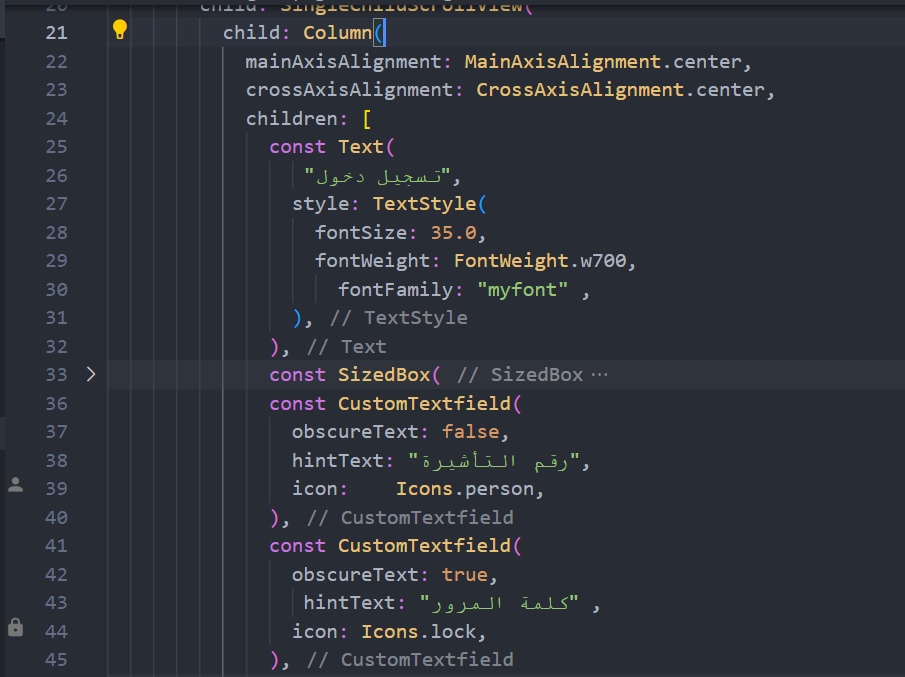


Figure 19 - Example of Flutter login

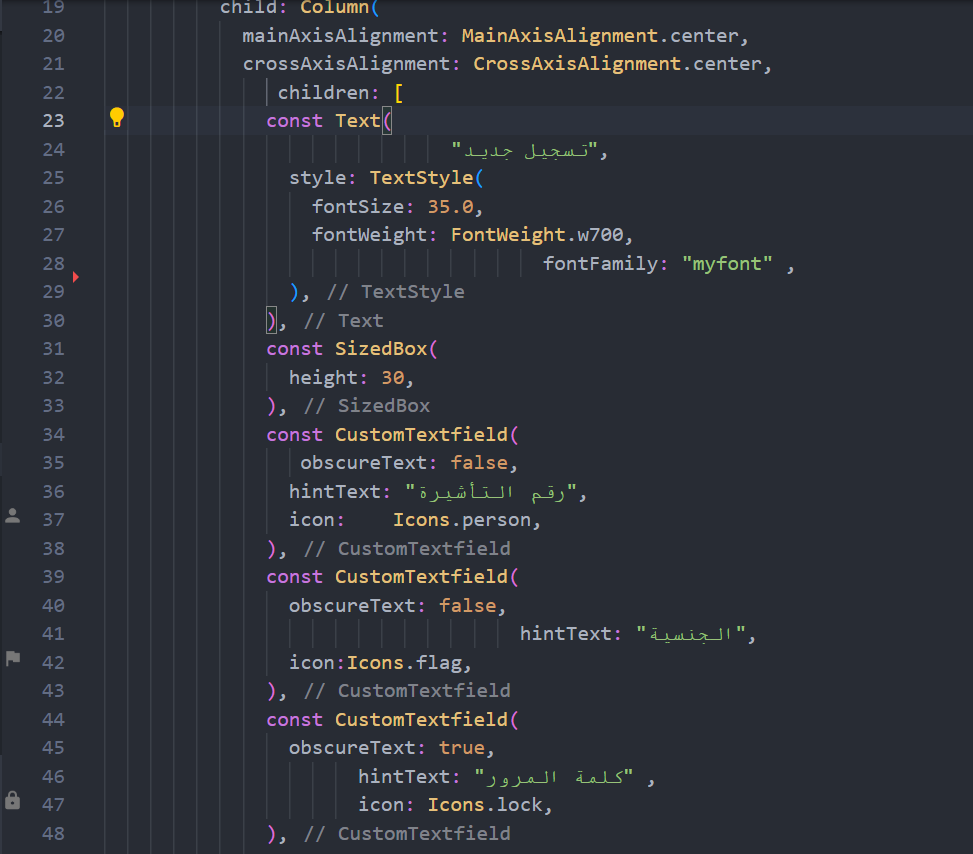


Figure 19 - Example of Flutter login

## Programming language

We used Flutter programming language for programming the application. We choose Flutter programming language based on some points such as:

* **Flutter is easy to learn**

Flutter was design to be easy to use and so easy to write, compile, debug, and learn rather than other programming languages.

* **Flutter is object-oriented**

This allows you to generate segmental programs and reusable code.

* **Flutter is platform-independent**

One of the greatest major advantages of Flutter is its ability to move simply from one computer system to another. The ability to run the same program on various different systems is critical to World Wide Web software, and Flutter succeeds at this through being platform-independent at both the source and binary levels.

* **Flutter is distributed**

Flutter is aimed to make distributed computing easy with the networking capability that is inherently combined into it.

* **Flutter is secure**

Flutter considers security as part of its design.

## Mapping from designing to Implementation

We mean by mapping designing to implementation transform diagrams to codes.

Each diagram in design phase implemented by coding it in programing language.

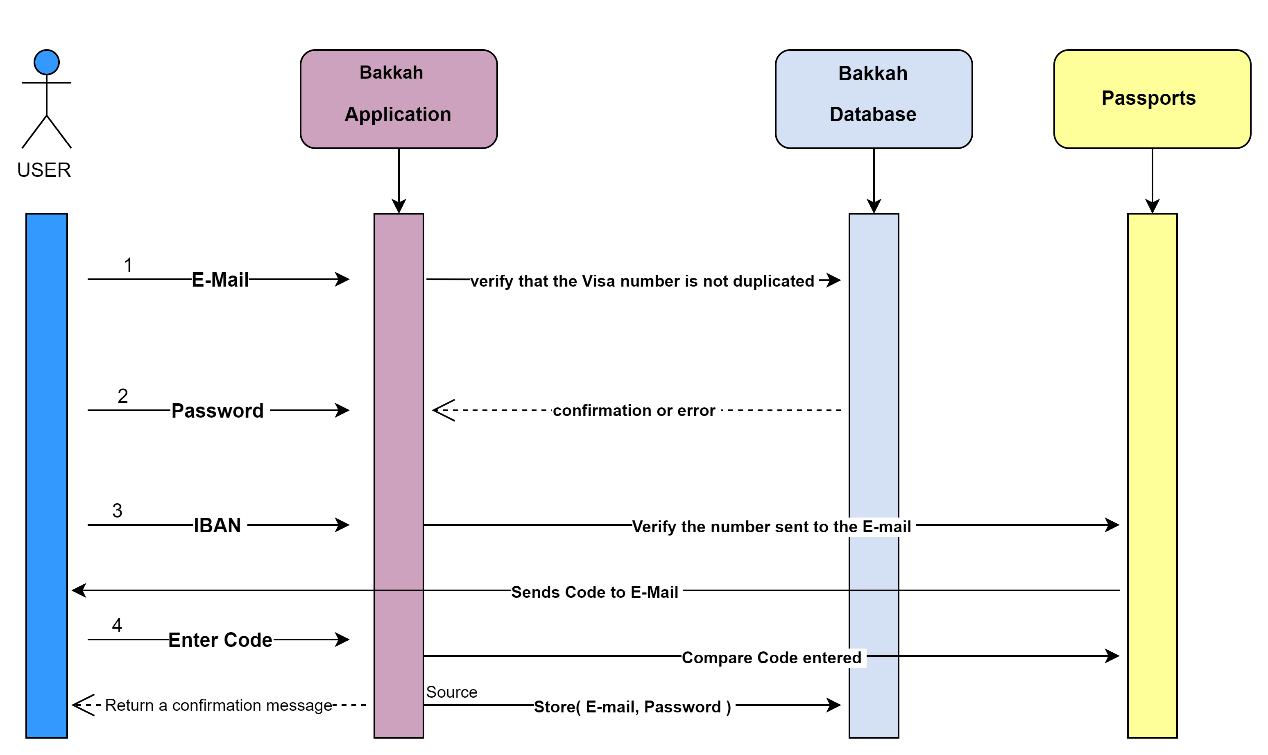


Figure20 - Register Sequence Diagram

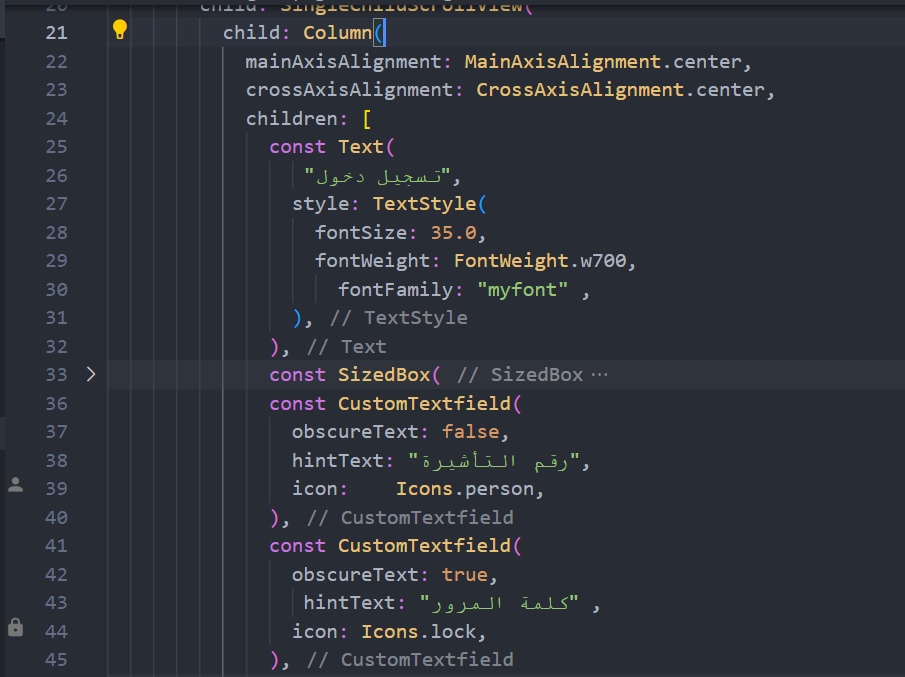


Figure 21 - The register operation converted to code

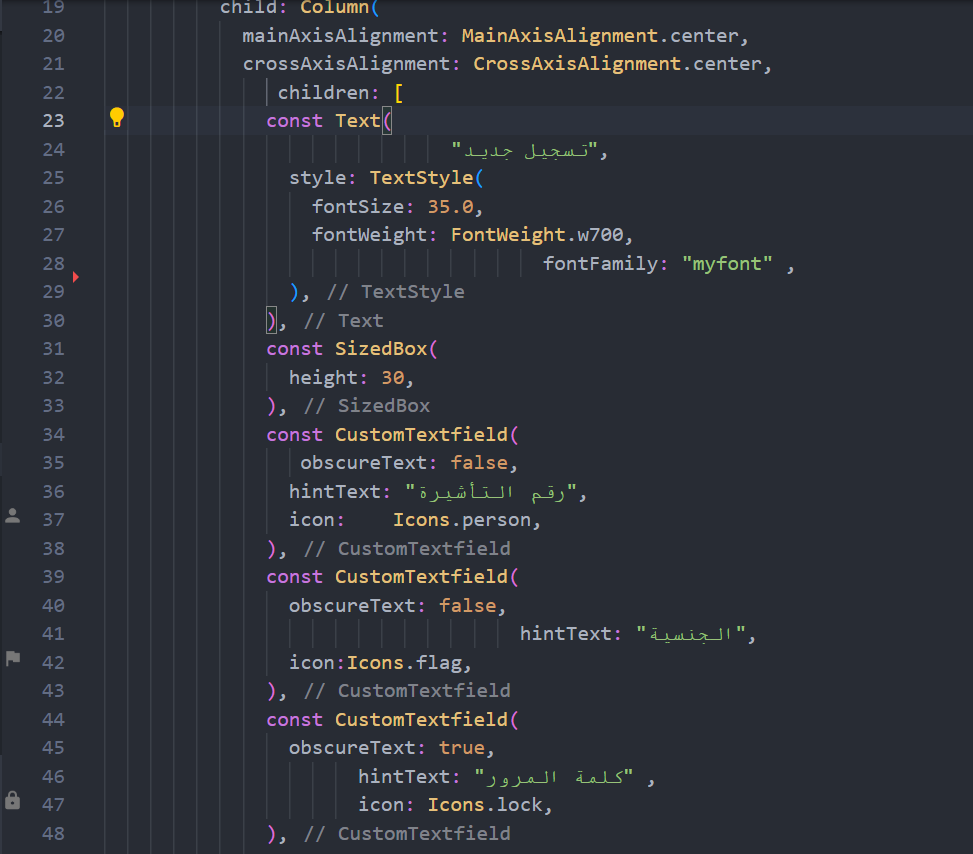


Figure 22- The register operation converted to code

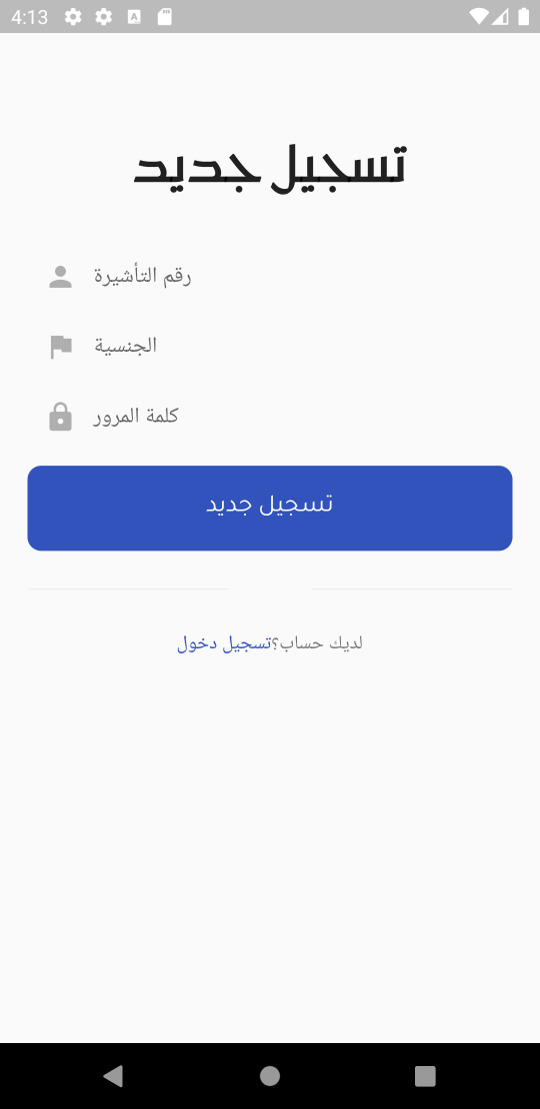


Figure 23 - Register interface

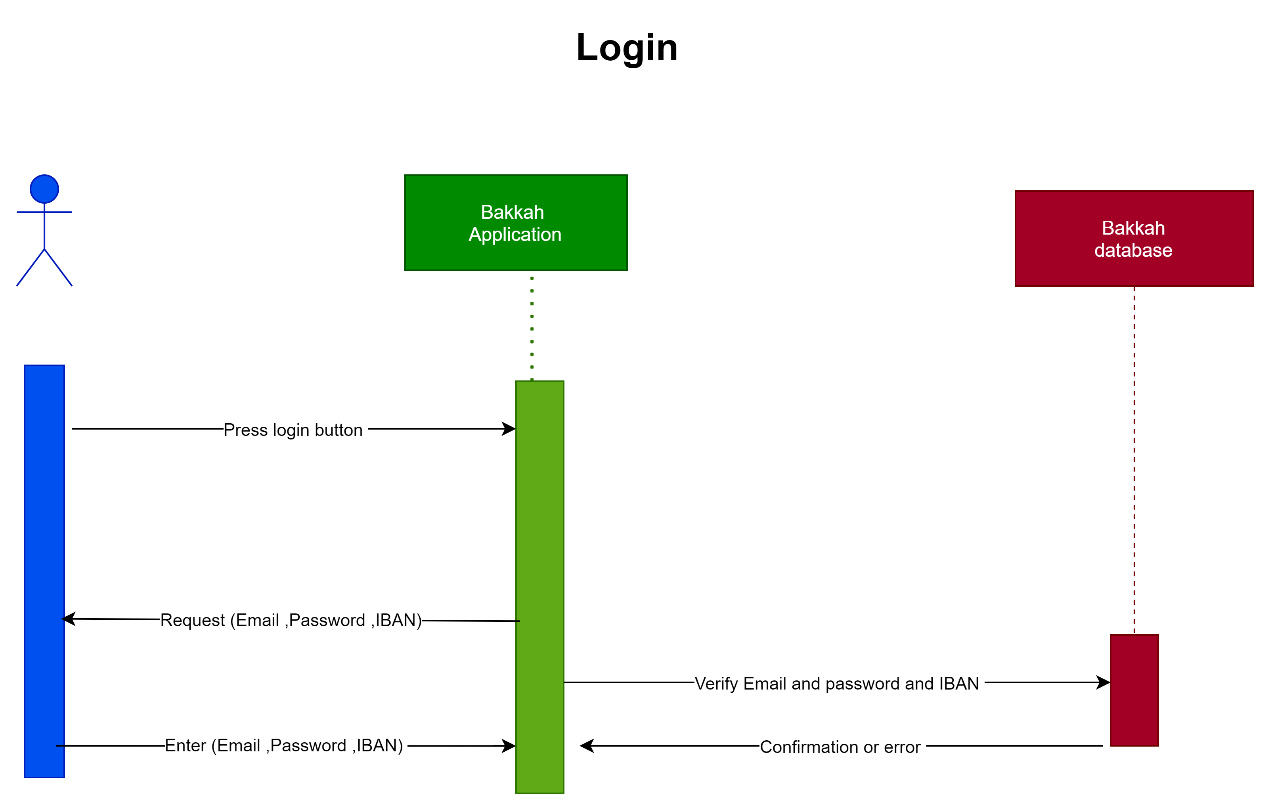


Figure 24- Login sequence diagram

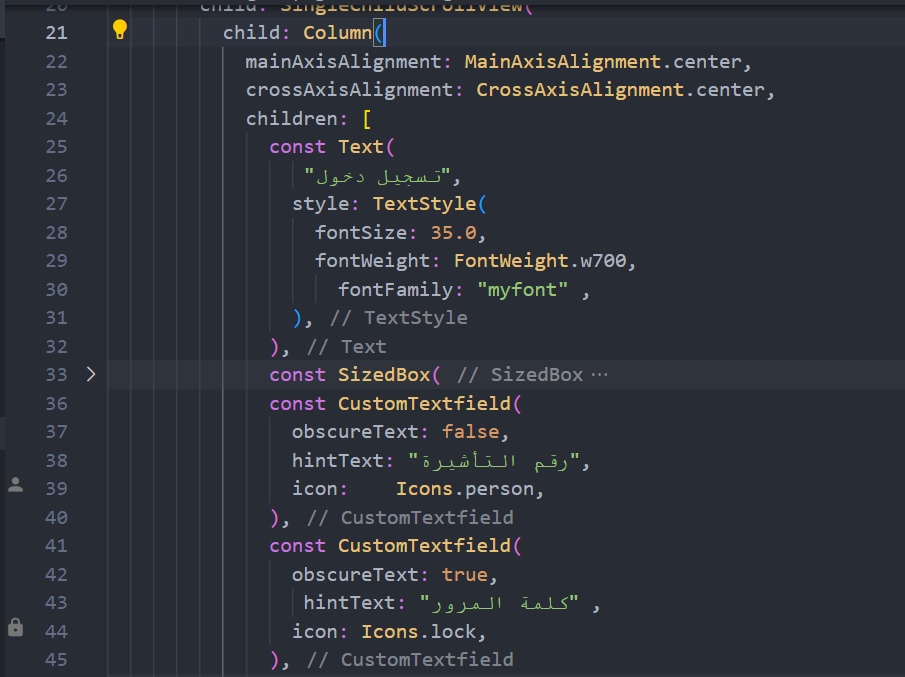


Figure25 - Login operation converted into code

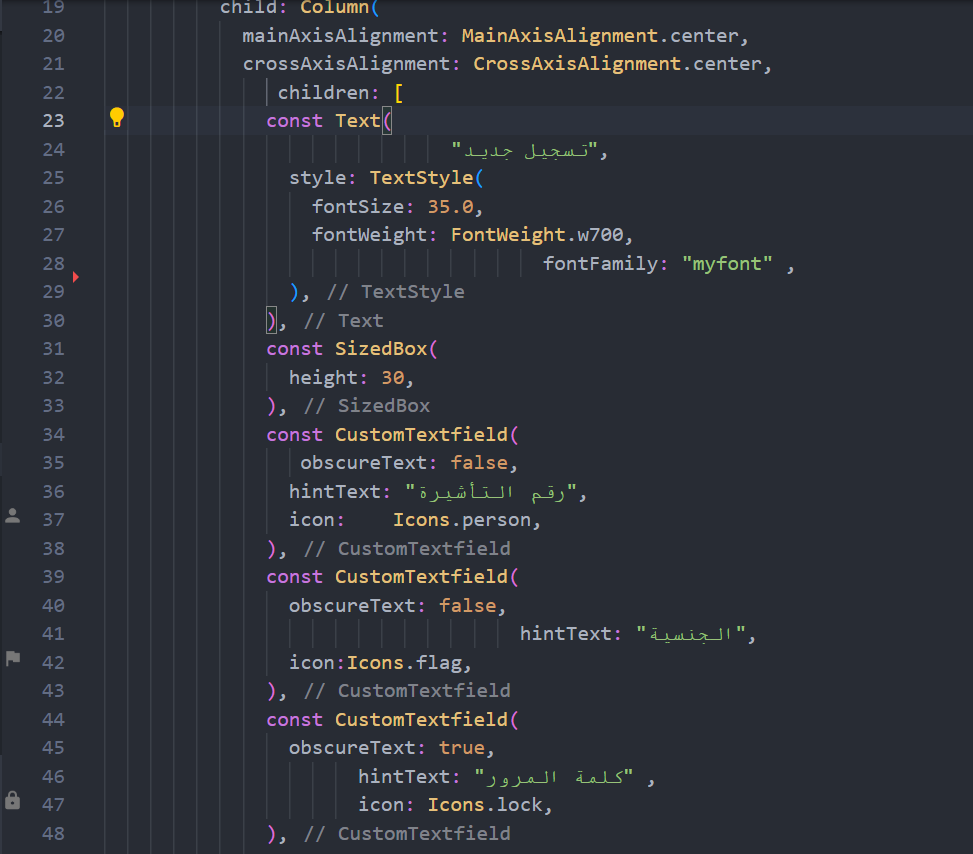


Figure 26 - Login operation converted into code

## Code Readability and Maintainability

* **Readability**

Readability is a human judgment of how the written text is easy to comprehend. The readability of a program is associated with its maintainability, and thus a key factor in overall software quality.

* **Maintainable**

Code has various features. In general, code is assumed to be maintainable when it is contains all of the following.

* **Understandable**

Anyone can pick up the code and figure out its purpose and general approach without a walkthrough by the original developer

* **Intuitive**

In the code just look to make sense, no matter how complex process.

* **Adaptable**

The code is written in such a way that alterations in data don’t require a complete rewrite.

* **Extendable**

Care has been set in the code architecture to allow extension of the essential functionality in the future.

* **Debuggable**

When something goes incorrect, the code provides enough information to detect the issues as directly as possible.

# Testing

## Introduction

Software testing is a process of implementing a program or application with the intent of finding the software bugs.

* It can also be specified as the process of validating and verifying that a software program or application or product:
* Meets the business and technical requirements that showed it’s design and development
* Works as predictable
* Can be executed with the same characteristic.

## Mobile application testing issues

Mobile application testing is different and more difficult than testing traditional desktop and has its own group of new challenges. The key challenges will be introduced in the following.

### Some of the issues

1. Screen scopes.
2. Connection types.
3. Different Android versions.
4. Power consumption and battery lifetime.
5. 5Security issues.
6. Usability.

## Testing Methodologies

Software testing methodologies are the different methods and techniques of confirming that a software application is completely tested. Software testing methodologies involve everything from unit testing individual modules, integration testing an entire system to specific forms of testing such as security and performance.

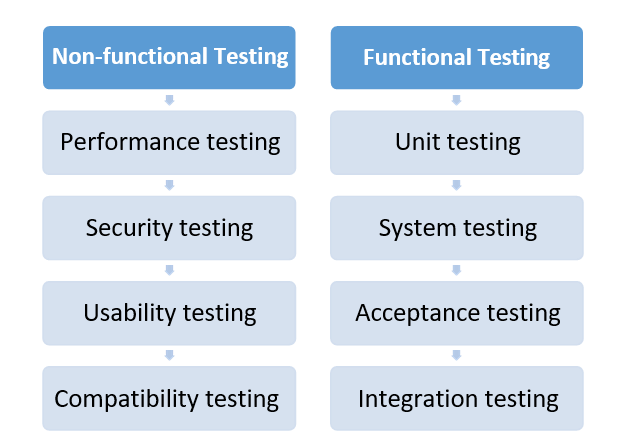


Figure 27- Testing Methods.

### Functional Testing

The functional testing portion of a testing methodology is usually broken down into four components - unit testing, integration testing, system testing and acceptance testing – commonly performed in this order. Each of them is defined below:

### Unit Testing

The Unit testing portion of a testing methodology is the testing of separate software modules or components that make up an application or system.

### Integration Testing

The Integration testing portion of a testing methodology is the testing of the different modules/components that have been successfully unit tested when combined together to accomplish specific tasks and activities.

### System Testing

The system testing portion of a testing methodology includes testing the entire system for errors and bugs. This test is carried out by interfacing the hardware and software modules of the entire system (that have been earlier unit tested and integration tested), and then testing it as a whole.

### Acceptance Testing

The acceptance testing portion of a testing methodology is the last phase of functional software testing, includes making sure that all the project requirements have been seen and the end-users and clients have verified the system to ensure it works as expected and meets all their defined requirements.

### Non-Functional Testing

Non-functional testing includes testing the application against the Non-functional requirements, which usually include testing the application against defined technical qualities, for example: weakness, scalability, usability. Some examples of non- functional testing are defined below:

### Performance, Load, Stress Testing

There are many types of performance testing in most testing methodologies, for example: performance testing is determining how a system works under an increasing load , load testing is validating that the system can work at the required response times once exposed to its predictable load , stress testing is finding the failure points in the system once the tested load exceeds that which it can support.

### Security, Vulnerability Testing

In the past, security was something that was verified after the fact. With the growth in cyber-crime and the awareness of the threats associated with software weaknesses, application security is now something that essentials to be designed and developed at the same time as the preferred business functionality. Security testing tests the software for confidentiality, integrity, authentication, availability, and non- repudiation. Separate tests are conducted to stop any unauthorized access to the software code.

### Usability Testing

The usability testing portion of a testing methodology looks at the end-user usability part of the software. The ease with which a user can entrée the product forms the main testing point. Usability testing looks at five phases of testing, - learnability, efficiency, satisfaction, memorability, and errors.

### Compatibility Testing

The compatibility portion of a testing methodology checks that the product or application is compatible with all the specified operating systems, hardware platforms, web browsers, mobile devices, and other designed third-party programs.

### Black Box Testing

The method of testing without having any knowledge of the inner workings of the application. The tester is unaware to the system architecture and does not have access to the source code.

### White Box Testing

White-box testing is the comprehensive investigation of inner logic and structure of the code. White-box testing is also called glass testing or open-box testing. In order to accomplish white-box testing on an application, a tester needs to know the inner workings of the code.

### Gray Box Testing

Grey-box testing is a method to exam the application with having a partial knowledge of the inner workings of an application. In software testing, the phrase the more you know, the better carries a lot of weight while testing an application.

## Test Plan

Test planning, the greatest important activity to make sure that there is initially a list of responsibilities and milestones in a baseline plan to track the movement of the project.

In our project we will concerned with two type of testing:

Functional testing: Concerned with the functional aspects of the M-Pharmacy to ensure meet the requirements.

1. Nonfunctional testing: Concerned with the nonfunctional requirements of M- Pharmacy to ensure meet the requirements.

### Functional Testing

Features to be tested

All the use cases appeared in system analysis document will be tested.

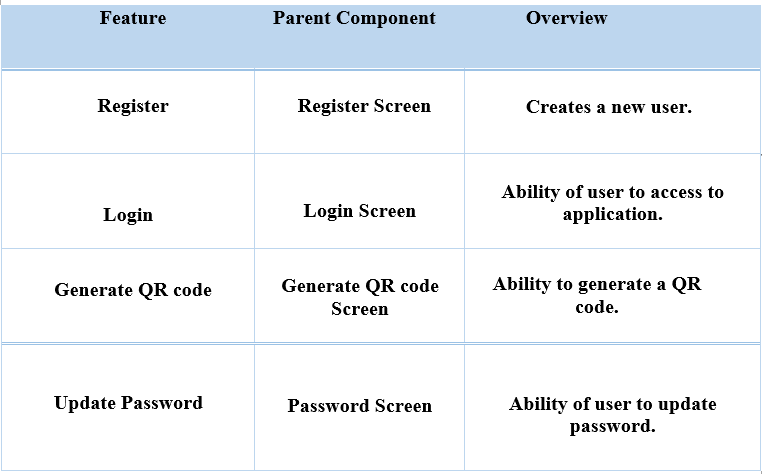


Figure 28- User Cases

## Testing tools, Data, Environment

**Testing tools**

Android provide various tools that are beneficial for test application. Such as:

**Robotium**

Robotium is one of the first and repeatedly used automated testing tools for software supported on Android. Robotium is a free Android UI testing tool. It is suitable for tests automation for different Android versions and, Software makers often define it as Selenium for Android. Tests created by Robotium are written in Flutter.

**MonkeyRunner**

MonkeyRunner is one of common Android Testing tools used for automating of functional tests for Android software. This tool is lower level than Robotium once does not have to deal with the source code in order to automate tests. The tests are written in Python.

**Ranorex**

Ranorex is a useful tool for tests automation not only for the newest, but also for early versions of android, start from Android 2.2. One of Ranorex benefits is its detailed reports with screenshots. It can link a smartphone or a tablet to Internet through Wi-Fi.

**Appium**

It’s a framework for making automated tests for iOS and Android. It is a free tool. It supports Android versions from 2.3 and later. Appium uses WebDriver interface for tests running. It supports many programming languages, such as Java, C#, Ruby and other which are in the WebDriver library.

**UI Automator**

This tool has been newly expounded by Google. It supports Android versions from 4.1 and later. One must choice another Android app testing tool in order to automate tests for earlier versions. UI Automator is capable to interact with all types of software products for Android, involving system applications. This allows UI Automator to lock and unlock a smartphone or a tablet.

### Testing environment

Test environment involves of elements that support test accomplishment with software, hardware and network configured. Test environment configuration must simulator the production environment in order to discover any environment/configuration associated issues.

### Factors for designing Test Environment

* Define if test environment needs archiving in order to take backups.
* Validate the network configuration.
* Determine the required server operating system, databases and other components.
* Determine the number of certificate required by the test team.

### Environmental needs

**Hardware**

* Laptop or Desktop PC.
* Mobile with Android platform.

### Software

* Operating System.
* Internet Browsers, Database management system.

### Others

* Internet connection for PC.
* Wi-Fi/4G/ connection for mobile.

## Test cases

A test case is a document which has a set of test data, prerequisites, expected results and post conditions, established for a particular test situation in order to validate compliance against a specific requirement. Test Case works as the beginning point for the test accomplishment, and after applying a set of input values, the application has a certain outcome and leaves the system at specific end point or also known as execution post condition.

### User Case

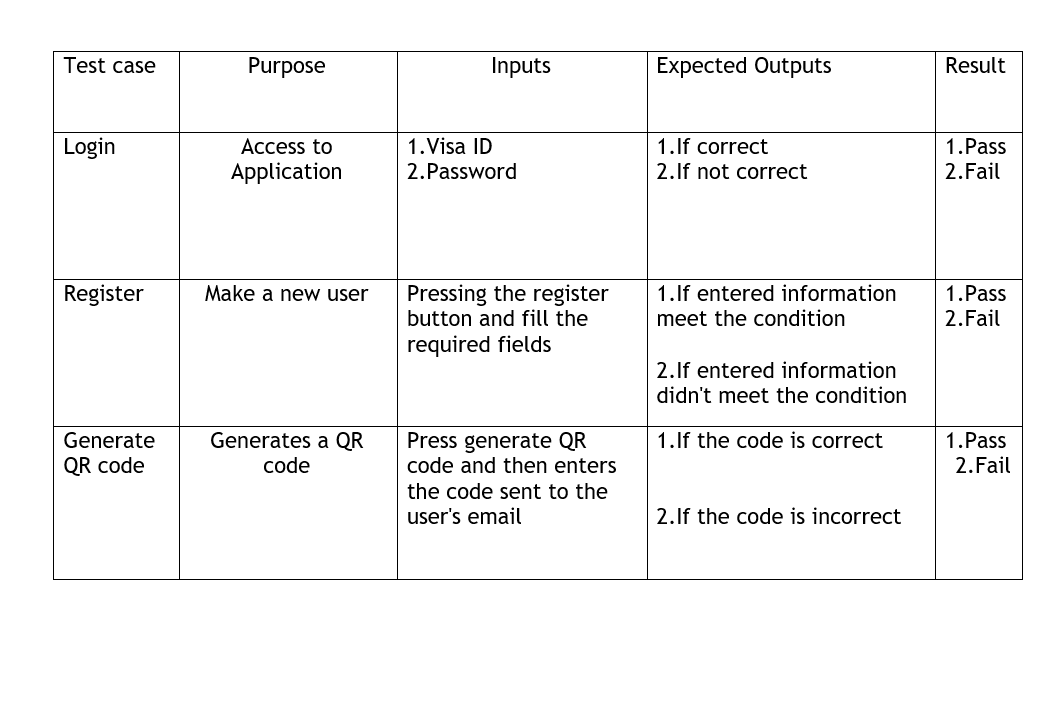


Figure 29- User case



Figure30 - First Screen

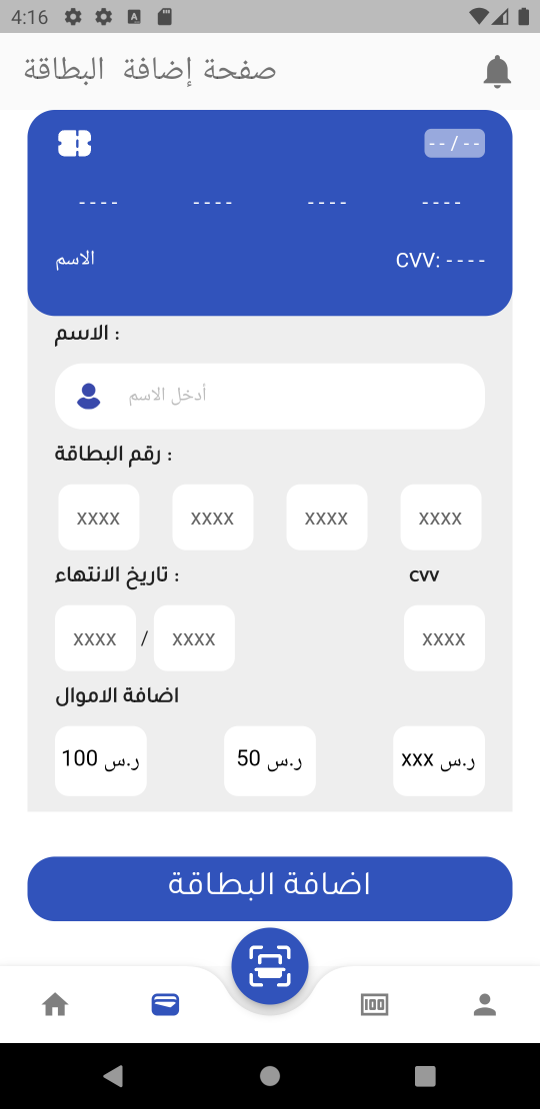


Figure 31 - Main screen

### Security testing

Security testing checks the software for confidentiality, integrity, authentication, availability, and non-repudiation.

## Conclusions

## Bakkah is a mobile application, so that the user can get his information securley, in this system, the application will provide a QR code to users when they need to transfer money The project gives us the opportunity to work on a real information system. A detailed analysis of the Bakkah project was carried out. An agile methodology was used to achieve rapid development. Bakkah system can take action from the user using user-friendly interfaces.

## Future Work

In the process of developing this project, we have learned many things and we are still learning as this project. Implementation remains ahead and one thing became very clear as we advanced in this project. There is no limitations to the extensions, technologies and features that can be added to this project. We hope that soon we will be able to implement this project on mobile devices with more features for more appropriate use.

## References

## <https://mockflow.com/>

## <https://www.diagrams.net/>

## <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>

## <https://vironit.com/functional-vs-non-functional-requirements/#close?llsknry=33029944>

## <https://www.lucidchart.com/pages/uml-class-diagram#:~:text=One%20of%20the%20more%20popular%20types%20in%20UML,must%20be%20present%20>

## <https://www.oreilly.com/library/view/learning-uml/0596003447/ch04.html>

## <https://www.microtool.de/en/knowledge-base/what-is-an-activity-diagram/>

## <https://creately.com/diagram/example/i92hdqkj/user-registration>

<https://www.lucidchart.com/pages/uml-sequence-diagram#:~:text=A%20sequence%20diagram%20is%20a%20type%20of%20interaction,new%20system%20or%20to%20document%20an%20existing%20process>.

<https://www.tutorialspoint.com/uml/uml_class_diagram.htm>

<https://www.smartdraw.com/entity-relationship-diagram/>

https://encyclopedia2.thefreedictionary.com/Structural+Model

<https://www.tutorialspoint.com/uml/uml_activity_diagram.htm>

<https://www.tutorialspoint.com/uml/uml_use_case_diagram.htm>

<https://toucan.com.jm/creating-an-effective-scope-of-work-for-a-mobile-application-or-website/>

<https://www.ninetwothree.co/blog/5-steps-for-outlining-the-scope-of-your-app-development-project>

<https://thedigitalprojectmanager.com/projects/pm-methodology/project-management-methodologies-made-simple/>

<https://www.projectmanager.com/blog/project-management-methodology>

<https://www.lucidchart.com/blog/how-to-draw-architectural-diagrams>

<https://learn.microsoft.com/en-us/message-analyzer/understanding-the-filtering-language-basics>

<https://learn.microsoft.com/en-us/message-analyzer/using-the-filtering-language>

<https://learn.microsoft.com/en-us/message-analyzer/procedures-using-the-data-filtering-features>

<https://learn.microsoft.com/en-us/message-analyzer/filtering-loaded-input-data>

<https://learn.microsoft.com/en-us/message-analyzer/filtering-captured-input-data>

<https://softwaretestingfundamentals.com/test-plan/>

<https://www.softwaretestinghelp.com/most-popular-web-application-testing-tools/>

<https://www.tutorialspoint.com/software_testing_dictionary/test_environment.htm>

https://www.softwaretestinghelp.com/software-compatibility-testing/

<https://www.youtube.com/watch?v=FVgQcfXHeGo>

## <https://www.youtube.com/watch?v=ExKYjqgswJg>

## <https://www.youtube.com/watch?v=kVMRzZjcLgM&t=67s>

## <https://www.youtube.com/watch?v=-_9Dsk51cq8>

## <https://www.youtube.com/watch?v=T4H7Z-MWQNI&t=13s>

## <https://www.youtube.com/watch?v=zDToEzwECDU>