International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 /doi:10.14445/22315381/IJETT-V69I5P206

Volume 69 Issue 5, 32-40, May 2021
© 2021 Seventh Sense Research Group®

A Software Engineering Approach in Design and Development of a Mobile Application

Minimol Anil Job[#1], S. Mustafa Alseba[*2]

[#1] *Assistant Professor, Arab Open University, Kingdom of Bahrain*

[#2] *System Analyst, Bahrain Petroleum Company, Kingdom of Bahrain*

[1]m.aniljob@aou.org.bh, [2]s.mustafa85@gmail.com

Abstract: *In the current technological era, mobile computing and the Internet of Things (IoT) have been used globally in various domains. Mobile technology is one of the fast-growing discipline and it is considerably important in professional and personal day-to-day lives of people. Smartphones have become part of the daily life and almost every daily tasks are done using mobile phones. This paper presents an innovative solution of donating used things with the help of a mobile application. The paper focuses on the requirement analysis, design, implementation and testing stages of the donation mobile application. The main aim of the development of this mobile app is to establish a connection between donor and others who would like to receive the items. This application is targeting the population of the Kingdom of Bahrain. This app aims to decrease the environmental impact and spread the spirit of solidarity in the society.*

Keywords: *mobile app, mobile computing, software engineering, cross-platform, requirement analysis*

I. INTRODUCTION

When people become more aware of the environmental risks of waste disposal, especially furniture and electronic equipment, and it is not legal to throw large objects in garbage cans as per the laws, discarding these items becomes difficult. Many people face difficulty in changing their home furniture or removing their home electronics equipment, as well as trying to discard some of their personal collectibles as well. Leaving electronic items may even affect the environment pollution as well. On the other hand, if the furniture or any equipment is free, there will be interested parties to take them. However, there is a need to find out a way to advertise it because there are no specific online platforms available for this.

Almost 2.22 million mobile connections are in the Kingdom of Bahrain in January 2021 as mentioned in the DIGITAL 2021 report. In one-year period, the number of mobile connections have been increased by 21 thousand (+1.0%). According to the report, the current mobile connections in the Kingdom of Bahrain are equivalent to 28.9% of the total population [23]. Mobile phone usage has become very popular among the people, the suggested solution for the donation problem is to develop a mobile app to provide an easy platform for donation. The donor mobile app is aimed to provide a platform to connect the donor with the requestor by ensuring privacy in order to make sure that the donor and the receiver information is protected. The app is similar to online stores and should be easy and straightforward. The mobile app is developed using a cross platform framework. The app targeting two types of users: public users (who searching for furniture and other equipment they want) and the Donor (who donate used furniture or equipment).

II. LITERATURE REVIEW

The most significant databases for academic journals in the area of Computer Science have been searched for conducting the review of related literature. Keywords related to the study; for instance, “Mobile application development”, “Software Engineering” and “Cross-Platform” are used for searching and acquiring results. Other tools and methods such as Boolean searches using AND (+), OR (|) and NOT (-) operators, phrase searches using quotation marks and wildcard searches using asterisks also used in getting results.

A mobile application also known as a mobile app in short is a software application that runs on smartphones, laptops, and other mobile devices. Remarkable advancement has been happened in the field of mobile application development during the past years [27]. Smartphones are becoming essential in people’s lives as a communication device because they offer multiple methods of collaboration using the various built-in sensor types [11]. Due to the availability of different types of operating systems, developers usually find difficulty in creating a standard application compatible for all of these. There are different types of mobile applications can be found such as native and hybrid applications. Native apps can only run on the smartphone operating system for which they were created. Each smartphone uses its own platform and programming language for development. Hybrid platform mobile apps offer components that make it easier to build applications that have a native feel to them and helps in saving time and cost as well [13]. The various mobile platform’s user interfaces are supported by its own design pattern.

A cross-platform uses common standardized web technologies. HTML, CSS, and JavaScript are used in a cross-platform and it is considered as an effective approach in the development of mobile applications [6]. Since they are standardized, widespread, relatively simple yet efficient, and well-supported, various web technologies are ideal for developing multi-platform applications. By combining additional features mobile devices' unique features can be prepared. The genuine need for the custom mobile application development introduced various mobile development methodologies and selecting the best methodology will make the development productive [7]. A book published by O'Reilly Media by an author Brian Fling explains the fundamental aspects of the mobile application development techniques to help the developers to create successful mobile applications [5].

In recent days, developers are focusing on cross-platform mobile app development. This type of development strategy supports an integrated application Programming Development (API) functioning on a native SDK. This will help building iOS and Android apps by sharing the same codebase. According to a published review paper, mobile applications developed by cross-platform can be made platform-independent. Some of the Native cross-platform applications are assembled using widely available frameworks such as Xamarin, React Native etc. [4]. Xamarin is the technology with a unique function cross-platform appeared in 2011. The main purpose of the Xamarin is to help the developers to design applications by reducing time. One of the advantages of this technology is that the developers are not required to spend time in coding similar applications for different platforms [14]. Many mobile application development frameworks are available due to today's growing demand and the growth in the use of mobile applications. They include native app development, cross-platform development frameworks, etc. Xamarin is a cross-platform mobile application development framework. Xamarin uses C# programming language to create hybrid apps that can run on different platforms. [10].

The mobile applications are used by people for various activities and these mobile apps or its notifications should provide information about the happenings around their social media circle. The developers should build a mobile application to deliver accurate, valid and quick information to the users [24]. For users, a well-organized mobile environment makes it easier to access the services in an efficient way [26]. The accessibility of mobile applications made them very popular and there is need for developers to look into the complexity. While developing the donation mobile application the developers considered the aspect of testing to ensure the correct functioning while routing to the market.

Users always expect mobile apps to be reliable and error-free. So, it is extremely important for application developers and software testers to ensure the quality of mobile applications before they release to market [1] [18]. New forms of mobile apps have arisen as smartphones have become more commonly used. The features of the mobile applications ca be used in online donation process. There are various types of mobile donation apps. These apps make donation process quick and easy without intervening users' daily lives, and automatically encourages donation process [2]. The review of the previously published papers has provided support in acquiring relevant information about various mobile application development practices including the cross-platform development platform. A cross-platform development tool is used in developing this donation mobile application.

The objective of presenting this paper is to demonstrate the systematic application of the software engineering approach in design and development of a user friendly donation mobile application using a cross -platform.

III. METHODOLOGY

For the development of the mobile app, the waterfall software development model is chosen because it is a sequential process move in a cascade mode, each step has concrete deliverables and is strictly documented. So, the developer cannot start the next step before the previous one is completed [3] [9] [17]. By searching in the different development models, it has been found that the most traditional software development model (waterfall model) is the best suited for the development of this mobile app [12] [18] [19]. The application has been designed by following the steps in Waterfall model: Requirement Analysis, Design, Implementation and testing.

In the classic waterfall process model, once a particular activity or phase is finished, it is not re-entered [22]. Also the activities do not overlap and follow a sequential lifecycle. The advantage is there is no need for reviewing or reworking. Software development life cycle can be applied in design and development of a good mobile application that meets the needs of users. Unified Modelling Language (UML) can be used to model the system which helps the developers during the stages of development [25].

A. Requirement and Analysis

Requirement Elicitation: The targeted users are people from Kingdom of Bahrain, the requirement elicitation was done through personal interviews [20]. Some of the interviewees provided few ideas about the privacy of the owner information.

The intended users of the system are guest, registered user and admin. A use case diagram demonstrating the use cases are depicted in Fig 1.

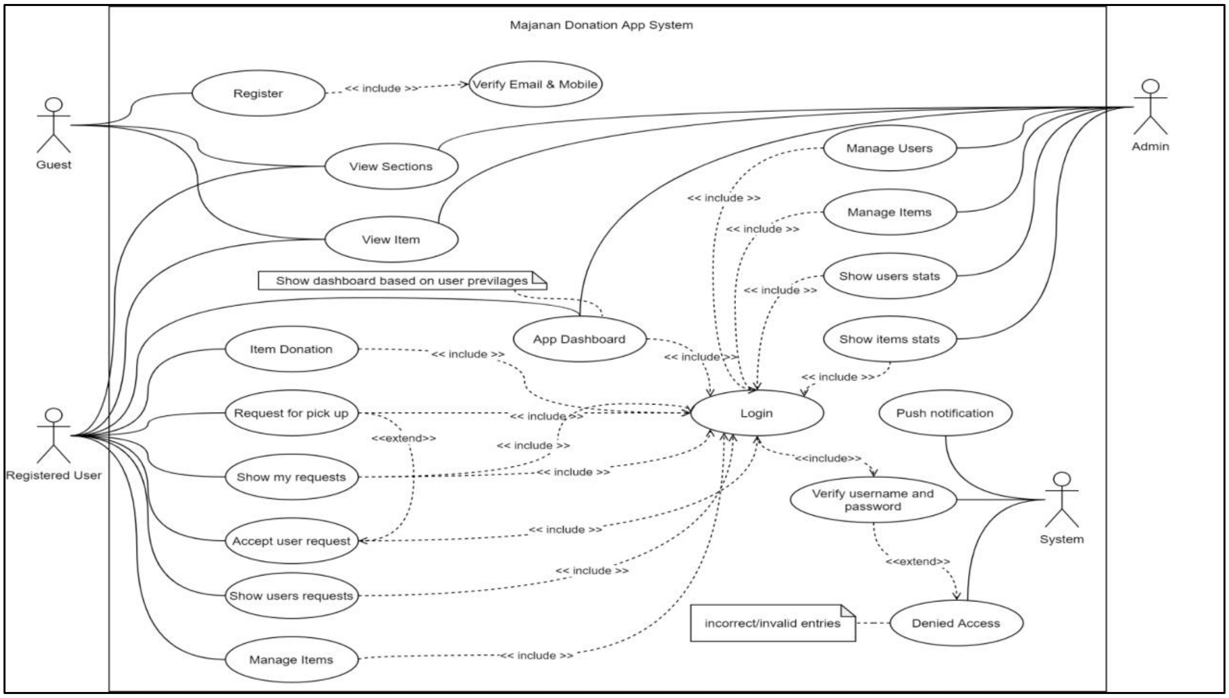


Fig 1: Use-Case diagram

As shown in the use case diagram, the actors of the system are guest, registered user and System Administrator. The system performs the authentication and push notification.

E-R Diagram

During the analysis stage, an Entity–relationship model (ER model) helps to demonstrate the structure of a database. The ER diagram is presented as a blueprint of the mobile application's database which will be implemented in the implementation stage. Fig 2 shows the ER diagram of the donation mobile app. The ER diagram is the conceptual data model and it shows the entities, attributes and the relationship between the entities.

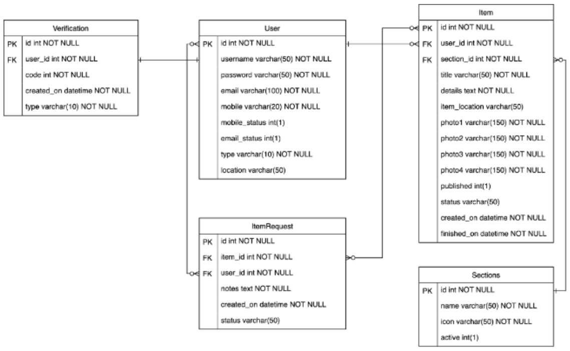

Verification
PK id int NOT NULL
FK user_id int NOT NULL
code int NOT NULL
created_on datetime NOT NULL
type varchar(10) NOT NULL
User
PK id int NOT NULL
username varchar(50) NOT NULL
password varchar(50) NOT NULL
email varchar(100) NOT NULL
mobile varchar(20) NOT NULL
mobile_status int(1)
email_status int(1)
type varchar(10) NOT NULL
location varchar(50)
Item
PK id int NOT NULL
FK user_id int NOT NULL
FK section_id int NOT NULL
title varchar(50) NOT NULL
details text NOT NULL
item_location varchar(50)
photo1 varchar(150) NOT NULL
photo2 varchar(150) NOT NULL
photo3 varchar(150) NOT NULL
photo4 varchar(150) NOT NULL
published int(1)
status varchar(50)
created_on datetime NOT NULL
finished_on datetime NOT NULL
ItemRequest
PK id int NOT NULL
FK item_id int NOT NULL
FK user_id int NOT NULL
notes text NOT NULL
created_on datetime NOT NULL
status varchar(50)
Sections
PK id int NOT NULL
name varchar(50) NOT NULL
icon varchar(50) NOT NULL
active int(1)

Fig 2: Entity Relationship diagram

B. Design

This stage in the development model explains the design techniques chosen for the development of the donation mobile app. Various diagrams used to model the system and they are presented below.

Component Diagram

This diagram shows the app API components, the diagram shows how the app interacts with API and AWS services work with each other, the diagram in Fig 3 also shows how the source code controlled.

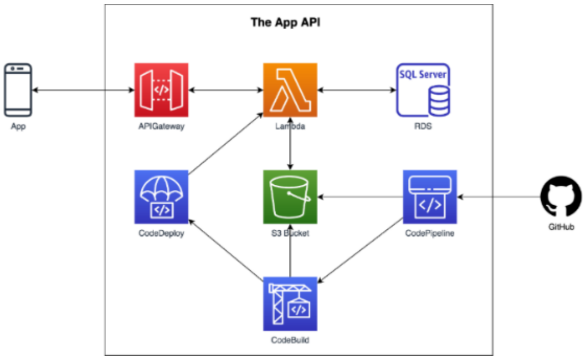


Fig 3: App API and source control

Data Flow Diagram (DFD)

The Data Flow Diagram (DFD) in Fig 4 is used to show a graphical representation of flow of data in the donation mobile app.

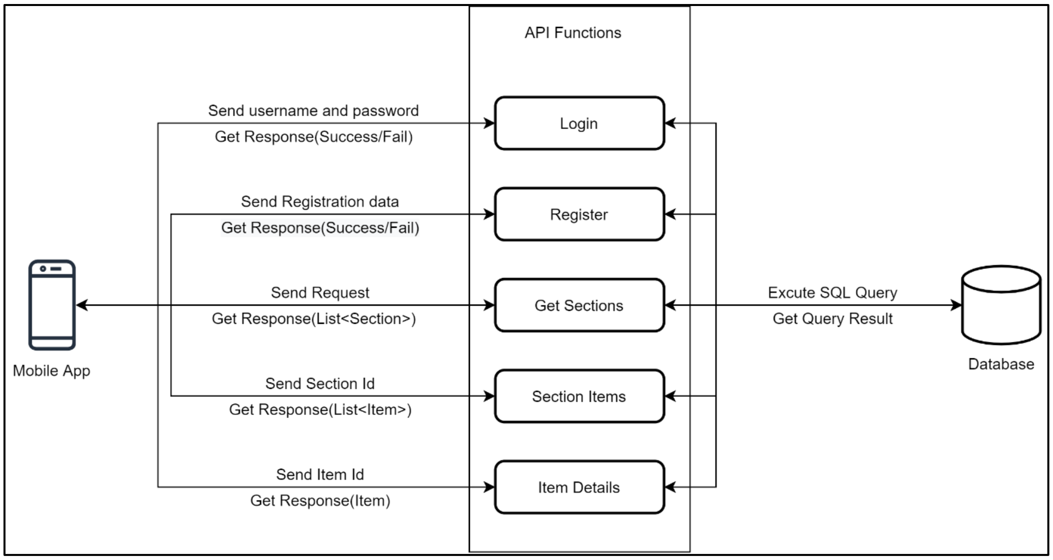
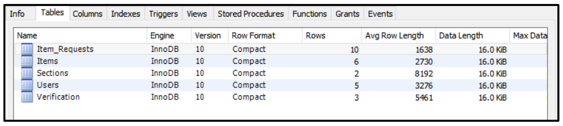


Fig 4: The application API data flow

C. Implementation

The implementation part is the most interesting area during the development because this part is converting the idea to a real mobile app. The implementation is carried in three stages: creating the database tables, developing the app API and finally, developing the mobile app. The implementation has done using the models identified in the conceptual model.

Database implementation: The database is the main part of the system which store all app data in it, MySQL is used as the database engine and MySQL Workbench to create and manage database and tables. MySQL is chosen for database engine because hosting it on MySQL much cheaper than MS SQL Server. Fig 5 shows a sample screen of database implementation.



Name	Engine	Version	Row Format	Rows	Avg Row Length	Data Length	Max Data
Item_Requests	InnoDB	10	Compact	10	1638	16.0 KiB	
Items	InnoDB	10	Compact	6	2730	16.0 KiB	
Sections	InnoDB	10	Compact	2	8192	16.0 KiB	
Users	InnoDB	10	Compact	5	3276	16.0 KiB	
Verification	InnoDB	10	Compact	3	5461	16.0 KiB	

Fig 5: implemented Database tables

Fig 6 shows the relational structure of the implemented database. All tables are implemented as specified in the ER diagram.

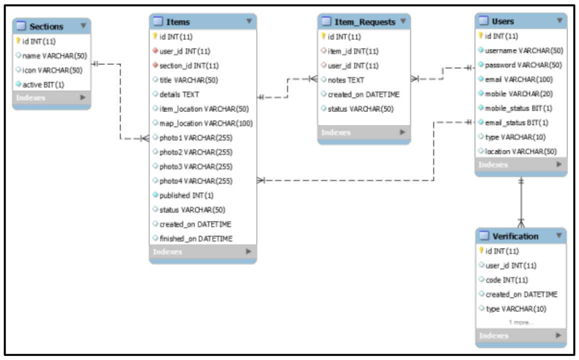


Fig 6: Implemented Database Schema

The implemented features in the donation mobile app are: Database connection, API Authentication, API calls (Login, Register, Load Items and others), App Base, API calls interface and App Screen (Login Page, Register Page, Items Page and others).

Application Program Interface (API): To develop the system, REST as API architectural and dot net core framework to develop API are used. One of the readymade MVC frameworks has been used because it makes developing API method easier. Dot net core is selected due to its compatibility with AWS Lambda. In addition, there is no need for any extra work to hosting it on a server less technology. Only one language, C#, for both App and API is used and sharing the code between them also possible. Fig 7 shows the login page of the donation mobile app

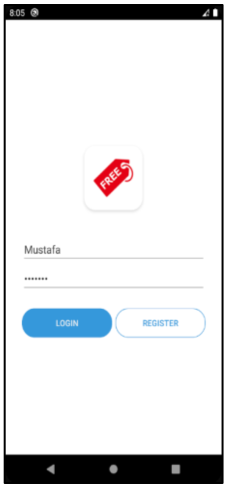


Fig 7: Screen shot of the login page

One of Xamarin Forms features is data binding, which provide best way to communicate between data and views. So, a new view model should be created which will request data from API and update the views.

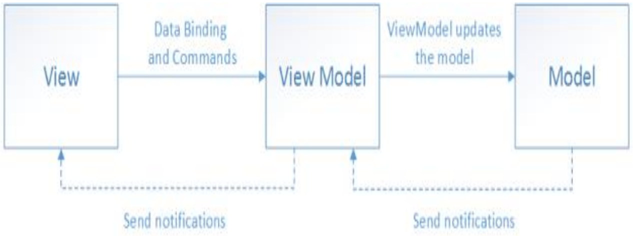


Fig 8:1 Data Binding in Xamarin [16]

Fig 9 and Fig 10 show samples pages from the donation mobile application

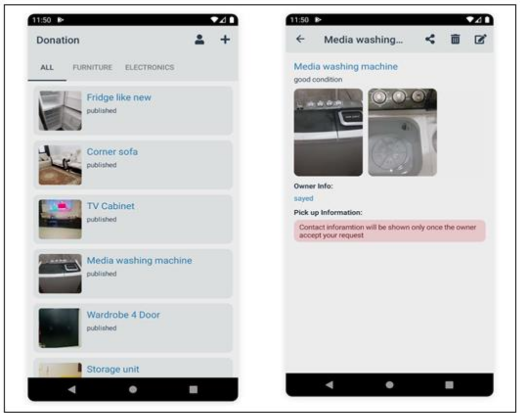


Fig 9:2 Samples of donation mobile app screens

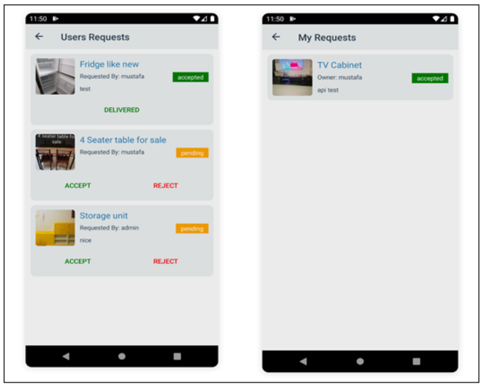


Fig 10:3 Samples of donation mobile app screens

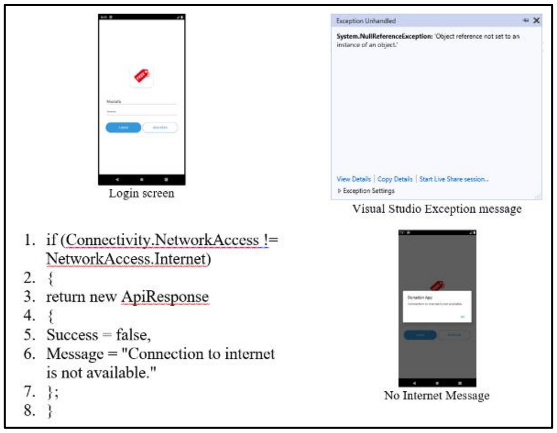
D. Testing

The classic waterfall process model suggests that testing is usually regarded as an activity that takes place at the end of software development [8]. The main target of any system is how it becomes a success and one of the success factors is avoiding error and bugs as much as possible because the first impression will determine if the user will continue using the system or not [15]. Testing is a part of the development cycle and this part will contain many processes to achieve it. Bothe white-box and black-box testing strategies are used to test the donation mobile app [21].

White-box Testing: is a type of testing that testing internal system function such as input validation.

Examples of white-box testing: No internet validation:

When try to login and device has no internet connection the app crash with because the HTTP request return stopped and return null value.

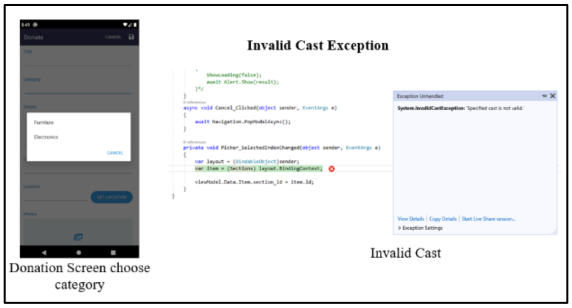

Login screen
Exception Unhandled
View Details | Copy Details | Start Live Share session...
Exception Settings
Visual Studio Exception message
Donation App
No Internet Message

System.NullReferenceException: 'Object reference not set to an instance of an object.'

```
if (Connectivity.NetworkAccess != NetworkAccess.Internet)
{
return new ApiResponse
{
Success = false,
Message = "Connection to internet is not available."
};
}
```

Fig 10: Sample Screen-1 White Box Testing

In "Donation Screen" the app crash on change category, invalid cast. To fix it just removed the line because is not needed any more.


Invalid Cast Exception

```
        ShowLoading(false);
        await Alert.Show(result);
    }*/
}
async void Cancel_Clicked(object sender, EventArgs e)
{
    await Navigation.PopModalAsync();
}

private void Picker_SelectedIndexChanged(object sender, EventArgs e)
{
    var layout = (BindableObject)sender;
    var item = (Sections) layout.BindingContext;

    viewModel.Data.Item.section_id = item.id;
}
}
```

Exception Unhandled

System.InvalidCastException: 'Specified cast is not valid.'

Donation Screen choose category

Invalid Cast

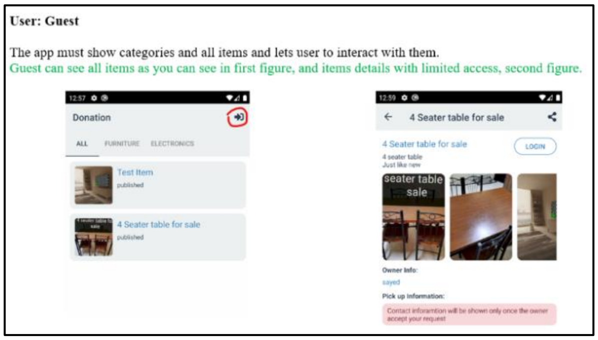
Fig 11: Sample Screen-2 White Box Testing

Black-box Testing

Black-box testing is a type of testing that done without knowing how internal functions works. The tester provides inputs and test the outputs of the system. The functional requirements are tested using black-box testing.

Functional Requirements

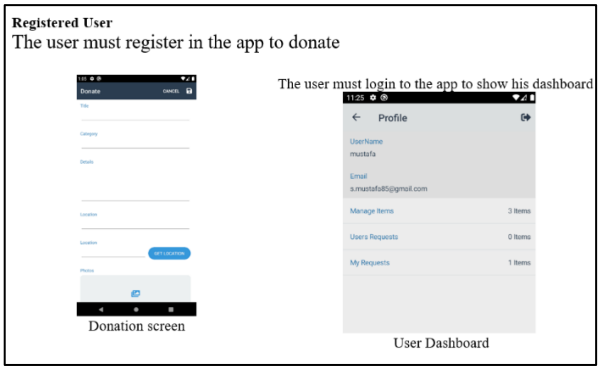
Two sample functional requirements test results are given below. The test cases are done for a guest user's and another one for a registered user's functionalities.


User: Guest

The app must show categories and all items and lets user to interact with them.

Guest can see all items as you can see in first figure, and items details with limited access, second figure.

Fig 12: Sample Screen-3 White Box testing


Registered User

The user must register in the app to donate

The user must login to the app to show his dashboard

Profile

UserName: mustafa

Email: s.mustafa85@gmail.com

Manage Items: 3 Items

Users Requests: 0 Items

My Requests: 1 Items

Donation screen

User Dashboard

Fig 13: Sample Screen-4 White Box testing

Non-Functional Requirements

A number of non-functional requirements (NFRs) are identified to ensure the quality of the mobile application during requirement analysis. These NFRs are tested after developing the mobile application. One of the NFR failed during the test is multi-language interface in both Arabic and English. Due to time constraint this NFR was not able to complete and it will be implemented in the next release of the application.

TABLE-1: NON-FUNCTIONAL REQUIREMENT TESTING

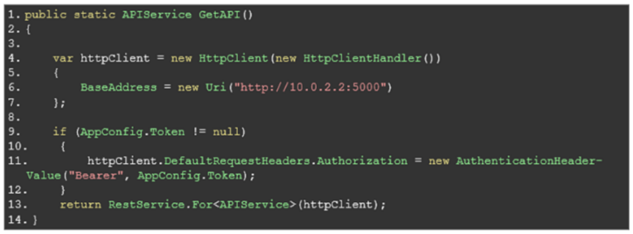
No.	Description	Type	Result
1	The app must be easy to use and user friendly. Test comment: • Easy to register a new user. • The user does minimum steps to do any function. • Is not complicated and straight forward.	Usability	PASSED
2	The user interface must be simple, clean, and beautiful. Test comment: • Design very simple and consistent with OS. • Used flat colors.	Look and feel	PASSED
3	The app should be fast and light. Test comment: The app size is very small just 17 megabytes when comparing it with the apps in the store.	Performance	PASSED
4	The app must have high availability. Test comment: The API works perfect and there was no downtime in the testing period.	Performance & Operational	PASSED
5	The app must be linked with server-side database through API.	Operational	PASSED
6	Not Registered users have only limited access to the app content.	Security	PASSED
7	The app should support English and Arabic languages Test comment: • This feature will be implemented in the next release because of the time limit.	Cultural	FAILD

IV. RESULTS AND DISCUSSION

Any success project should achieve at least the main objectives of the project and to achieve these project objectives should be placed in a realistic project plan. A realistic project plan was developed prior to the start of the development of the mobile app. Proper identification of the relevant tools and technologies made the development process smooth and successful. The proposed donation App became a real app by implementing and testing almost all features and requirements. One of the success factor to be mentioned here is the use of a cross-platform framework (Xamarin) and it helped in reducing development time. The framework also helped to produce the same output of using native tools. The app now runs of iOS and Android and adding any new feature is possible for both OS at the same time.

As mentioned earlier, the mobile application has been developed by using a cross-platform called Xamarin it is an open-source platform that allowed developers to develop one app to works on both iOS and Android.

***Development of API calls**:* After selecting NuGet, installed Refit and this library makes HTTP request much easier. First, created the API.cs class which contains all API services, then created an initialize method. Any service in the API can be called form this method. Authorization header token will be assigned if user is logged in.



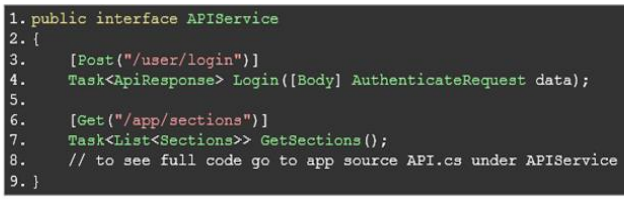
```
public static APIService GetAPI()
{

    var httpClient = new HttpClient(new HttpClientHandler())
    {
        BaseAddress = new Uri("http://10.0.2.2:5000")
    };

    if (AppConfig.Token != null)
     {
         httpClient.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", AppConfig.Token);
     }
     return RestService.For<APIService>(httpClient);
}
```

Fig 14: API Call method

The services interface is developed and each API method is written in this interface.

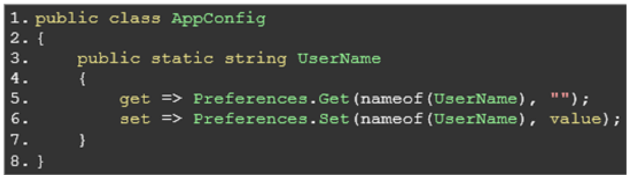


```
public interface APIService
{
    [Post("/user/login")]
    Task<ApiResponse> Login([Body] AuthenticateRequest data);

    [Get("/app/sections")]
    Task<List<Sections>> GetSections();
    // to see full code go to app source API.cs under APIService
}
```

Fig 15: API services interface

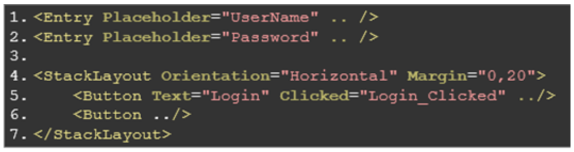
User Settings: Developing the user settings class, after selecting the NuGet and installed Xamarin.Essentials. This library provides easy access for cross-platform API by using one code, then create AppConfig.cs, this class holds all user setting to allow the developer to save and retrieve user setting easily.



```
public class AppConfig
{
    public static string UserName
    {
        get => Preferences.Get(nameof(UserName), "");
        set => Preferences.Set(nameof(UserName), value);
    }
}
```

Fig 16: AppConfig class

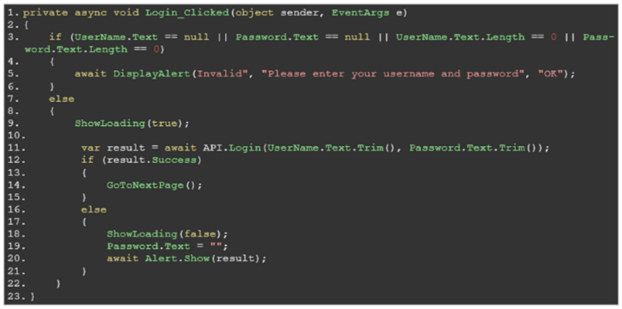
Login Page: Pages in Xamarin made from two parts XAML file (Design) and cs file (Code). The Login_Clicked method has been implemented and the API service can be called easily since it already implemented in API part.



```
<Entry Placeholder="UserName" .. />
<Entry Placeholder="Password" .. />

<StackLayout Orientation="Horizontal" Margin="0,20">
    <Button Text="Login" Clicked="Login_Clicked" ../>
    <Button ../>
</StackLayout>
```

Fig 17: Login Page XAML

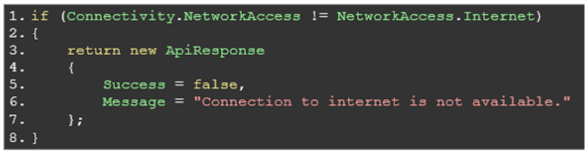


```
private async void Login_Clicked(object sender, EventArgs e)
{
    if (UserName.Text == null || Password.Text == null || UserName.Text.Length == 0 || Password.Text.Length == 0)
    {
        await DisplayAlert(Invalid", "Please enter your username and password", "OK");
    }
    else
    {
        ShowLoading(true);

         var result = await API.Login(UserName.Text.Trim(), Password.Text.Trim());
         if (result.Success)
         {
             GoToNextPage();
         }
         else
         {
             ShowLoading(false);
             Password.Text = "";
             await Alert.Show(result);
         }
     }
}
```

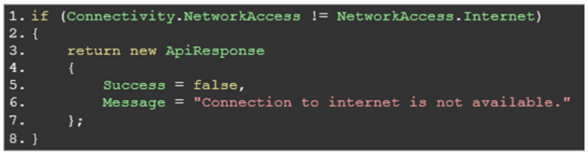
Fig 18: Login Button Code

Once login method called, it will check if the user has internet connection and i



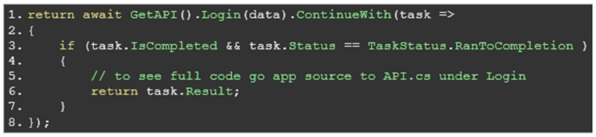
```
if (Connectivity.NetworkAccess != NetworkAccess.Internet)
{
    return new ApiResponse
    {
        Success = false,
        Message = "Connection to internet is not available."
    };
}
```

Fig 19:f not, the function will be stopped and returned the error to the main view. If internet is available, the app will execute login request and wait for response. If login succeeded, the user information will be saved in user data storage.



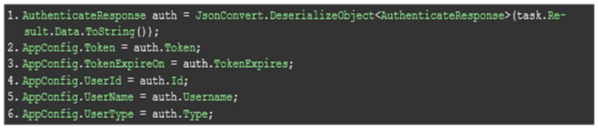
```
if (Connectivity.NetworkAccess != NetworkAccess.Internet)
{
    return new ApiResponse
    {
        Success = false,
        Message = "Connection to internet is not available."
    };
}
```

Fig 19: Check internet connectivity



```
return await GetAPI().Login(data).ContinueWith(task =>
{
    if (task.IsCompleted && task.Status == TaskStatus.RanToCompletion )
    {
        // to see full code go app source to API.cs under Login
        return task.Result;
    }
});
```

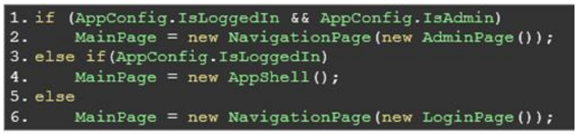
Fig 20: Execute login request



```
AuthenticateResponse auth = JsonConvert.DeserializeObject<AuthenticateResponse>(task.Result.Data.ToString());
AppConfig.Token = auth.Token;
AppConfig.TokenExpireOn = auth.TokenExpires;
AppConfig.UserId = auth.Id;
AppConfig.UserName = auth.Username;
AppConfig.UserType = auth.Type;
```

Fig 21: Save login information

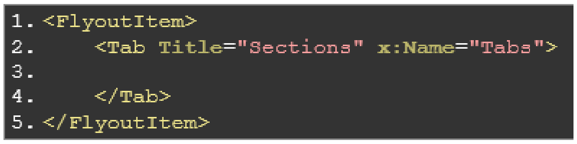
Main App Screen: After implementing login page, using App.cs and App() method allow the app to choose the main screen based on three factors, first, if the user logged the app will check user type and for admin users the main page will be AdminPage. For normal users the main page will be AppShell, otherwise the app will show LoginPage.



```
if (AppConfig.IsLoggedIn && AppConfig.IsAdmin)
    MainPage = new NavigationPage(new AdminPage());
else if(AppConfig.IsLoggedIn)
    MainPage = new AppShell();
else
    MainPage = new NavigationPage(new LoginPage());
```

Fig 22: Set main page based on user type

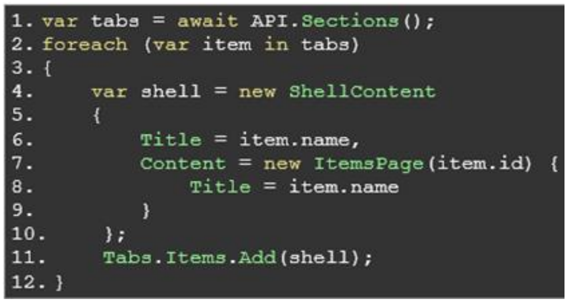
App Shell: Shell is one of helpful component in Xamarin which allow developer to create dynamic layout, such as flyout menu, tabbar, etc. in this app will use it to create top tabs to present sections tabs.



```
<FlyoutItem>
    <Tab Title="Sections" x:Name="Tabs">

    </Tab>
</FlyoutItem>
```

Fig 23: App Shell Layout

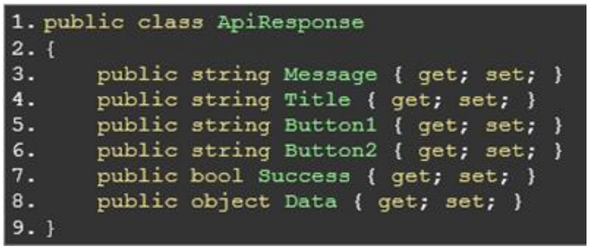


```
var tabs = await API.Sections();
foreach (var item in tabs)
{
    var shell = new ShellContent
    {
        Title = item.name,
        Content = new ItemsPage(item.id) {
            Title = item.name
        }
    };
    Tabs.Items.Add(shell);
}
```

Fig 24: Retrieve app sections from the server

Utilities Functions: In Donation app some utilities function has been created to simplify some features.

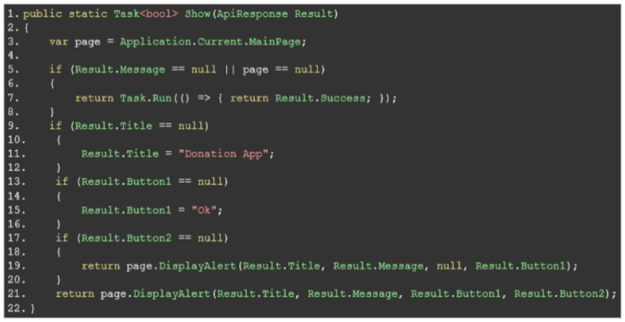
Alert Show: The API developed to response messages in one structure:



```
public class ApiResponse
{
    public string Message { get; set; }
    public string Title { get; set; }
    public string Button1 { get; set; }
    public string Button2 { get; set; }
    public bool Success { get; set; }
    public object Data { get; set; }
}
```

Fig 25: Implement response model

And to presenting these data in easiest for the user, Alert function is created in the app to take API result and showing it to the user.

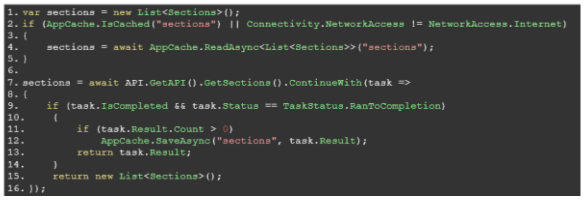


```
public static Task<bool> Show(ApiResponse Result)
{
    var page = Application.Current.MainPage;

    if (Result.Message == null || page == null)
    {
        return Task.Run(() => { return Result.Success; });
    }
    if (Result.Title == null)
     {
         Result.Title = "Donation App";
     }
     if (Result.Button1 == null)
     {
         Result.Button1 = "Ok";
     }
     if (Result.Button2 == null)
     {
         return page.DisplayAlert(Result.Title, Result.Message, null, Result.Button1);
     }
     return page.DisplayAlert(Result.Title, Result.Message, Result.Button1, Result.Button2);
}
```

Fig 26: Alert function

App Cache: Data caching is required to increase app performance, so App Cache is created to achieve this, now how is it work:



```
var sections = new List<Sections>();
if (AppCache.IsCached("sections") || Connectivity.NetworkAccess != NetworkAccess.Internet)
{
    sections = await AppCache.ReadAsync<List<Sections>>("sections");
}

sections = await API.GetAPI().GetSections().ContinueWith(task =>
{
    if (task.IsCompleted && task.Status == TaskStatus.RanToCompletion)
     {
         if (task.Result.Count > 0)
             AppCache.SaveAsync("sections", task.Result);
         return task.Result;
     }
     return new List<Sections>();
});
```

Fig 27: Working with App Cache

Donation using a mobile application is the best solution for the problem identified in the analysis phase since using mobile apps to daily tasks is the trend among people everywhere. The developed mobile application is fulfilling almost all the features that the user wants, especially the feature for safe connection between donor and requester. The mobile app was developed using open-source platform which provides efficient tools and libraries to develop the app features. After the implementation proper testing has been done to ensure the functioning of the implanted features.

Next step after the development of the mobile application is route to market. Therefore, it is too early to confirm that all objectives are achieved because some of the objectives should be tested only after releasing the app and when it is operational. Checking the user interface should be done with actual user experience. Performance test can be done using an actual stress test. It is important to mention that some of the requirements are not implemented yet due to financial aspects, such as automatic deployment and automating SMS to the users. It is worth mention that development of mobile application is not difficult because of the various resource available online. Various open-source strong and powerful tools and libraries are available to support software developers. Identifying an appropriate software development approach in any software project is essential.

The traditional waterfall model is best suited for the development of donation mobile app. This model is suggested for small project with a specific completion date. All development tools that used were open-source technologies; Visual studio community edition the main development IDE, Adobe XD which is used to create the application prototypes, *MySQL workbench* to manage database, *diagrams.net* for modelling the diagrams such as use case, ER, *Xamarin* for the application framework, and .*Net Core* for the API framework. Most of all, there is always an opportunity to help the society and create an impact in people's life.

V. CONCLUSIONS AND FURTHER WORK

In this paper, the researches presented the software engineering steps in the development of a donation mobile application. Even though it is worth sharing the development experience during this rapid application development era, the main goal of this paper is to present the experience of utilizing mobile application development methodologies by applying software engineering strategy in a systematic way. The classis waterfall model adopted in the development of the donation mobile application helped to arrange the activities of requirements, design, implementation and testing into a single sequence. The paper also demonstrates the success factor of a cross-platform development. Since the same set of codes is used by all platforms, cross platform software development can ensure the overall look of the app can be preserved. The advantage of cross-development platforms is that it helps in developing mobile apps by reducing cost and saving time.

Currently, the app focusing on individuals which makes a safe connection between donor and requester. In future, the app could provide a facility to allow local charities to create their own page to accept the donation from the users. Mainly, three features, (i) Adding change password feature, (ii) Support Arabic language and (iii)Verify the user mobile number by SMS verification are not achieved in the current app and it will be included in the next release.

REFERENCES

[1] Ashishdeep, A., Bhatia, J., & Varma, K., Software process models for mobile application development: A review, IJCSC, 7(1)(2015) 150-153, DOI: 10.090592/IJCSC.2016.020.

[2] Boreum Choi,Donation via Mobile Applications: A Study of the Factors Affecting Mobile Donation Application Use, International Journal of Human–Computer Interaction, 32(12).

[3] Chandra, V., Comparison between Various Software Development Methodologies, International Journal of Computer Applications, 131(9)(2015) 7-10, Retrieved May 25th, 2018,

[4] Chaitanya Kaul1 Saurav Verma2, A Review Paper on Cross Platform Mobile Application Development IDE, IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278-8727, 17(1)(2015) 30-33 Ver. VI www.iosrjournals.org

[5] Fling, B.: Mobile Design and Development: Practical Concepts and Techniques for Creating Mobile Sites and Web Apps, (2009) 13–27. O'Reilly Media ISBN 978-0-59615544-5

[6] G. Chen and D. Kotz, "A Survey of Context-Aware Mobile Computing Research," Hanover, NH, USA, Tech. Rep., (2000).

[7] H. Heitkötter, S. Hanschke and T. A. Majchrzak., Comparing Cross-platform Development Approaches for Mobile Applications, in Proceedings of the 8th International Conference on Web Information Systems and Technologies (WEBIST), Porto, Portugal, (2012).

[8] H. Heitkötter, S. Hanschke and T. A. Majchrzak, "Evaluating Cross-Platform Development Approaches for Mobile Applications, Web Information Systems and Technologies, (2013) 120-138.

[9] Hanif , S.Jagadeesan , Vinayak A., Priyanka C. Bhatt, Mobile Application Development: A comprehensive and systematic literature review, Proceedings of the International Conference on Industrial Engineering and Operations Management Pilsen, Czech Republic, (2019) 23-26.

[10] Hooda I, Chhillar RS. Software test process, testing types and techniques. International Journal of Computer Applications. Jan 1; (2015) 111(13).

[11] Ivar Jacobson, Harold "Bud" Lawson, Pan-Wei Ng, The Essentials of Modern Software Engineering: Free the Practices from the Method Prisons, ACM Books., (2019).

[12] K. Vishal and A. Kushwaha., Mobile Application Development Research Based on Xamarin Platform., in 2018 4th International Conference on Computing Sciences (ICCS), Jalandhar, India, (2018) 115-118. doi: 10.1109/ICCS.2018.00027

[13] Lane, N.D., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., Campbell, A.T.: A Survey of Mobile Phone Sensing. IEEE Communication Magazine 48(9)(2010) 140–150.

[14] Krishna, S.T, Sreekanth, S., Perumal, K., & Reddy, K.R.K., Explore 10 different types of software development process model. International Journal of Computer Science and Information Technologies, 3(4)(2012) 4580-4584.

[15] M. Kurosu (Ed.): Human-Computer Interaction, Part II, HCII 2013, LNCS 8005, (2013) 189–196. © Springer-Verlag Berlin Heidelberg 2013

[16] Mandanna B J, Dr. Suchitra R, 2016, Xamarin, International Journal of Engineering Research & Technology (Ijert) Ncrit 4(27) (2016).

[17] Meriem Atifi, Abdelaziz Mamouni, Abdelaziz Marzak, A Comparative Study of Software Testing Techniques, International Conference on Networked Systems, NETYS Networked Systems (2017) 373-390.

[18] Microsoft. (n.d.). The MVVM Pattern. Retrieved from Microsoft Documentation: https://docs.microsoft.com/en-us/xamarin/xamarin-forms/enterprise-application-patterns/mvvm

[19] Sommerville, I.: Software Engineering, 7th edn. Pearson Eductation Ltd., London (2004).

[20] X.Zou, Method of Construction: Modern Software Engineering, Posts & Telecom Press, (2018).

[21] Zhifang Liu,Xiaopeng Gao, Xiang Long., Adaptive Random Testing of Mobile Application",Computer Engineering and Technology (ICCET), 2010 2nd International Conference on 2(2010)V2-297 - V2-301.

[22] Zhang, H.; Babar, M.A. Systematic reviews in software engineering: An empirical investigation. Inf. Softw. Technol. (2013) 55, 1341–1354.

[23] https://datareportal.com/reports/digital-2021-Bahrain

[24] Jigar K Patel Mobile ERP: Implementation and Sustainment Strategies., International Journal of Computer Trends and Technology 68.11(2020) 10-14.

[25] Sariyun Naja Anwar, Isworo Nugroho, Rina Candra Noor Santi, Edy Supriyanto. Web Service on Andriod Mobile Platform in Application of Tourism Infrastructure Information. International Journal of Computer Trends and Technology, 69(3) 74-78

[26] Hegde Sharaj Bhaskar Shyamala, Krishnamoorthy Rao, Padmanabha Bhandarkar,Prateek Prakash Vetekar, "Survey On An Android Application for Home Services" SSRG International Journal of Computer Science and Engineering 6.11(2019) 60-63.

[27] Ruqiya, Noman Islam, Athaul Rai, Noman Khan. Usability Analysis of Android and iOS Operating Systems International Journal of Engineering Trends and Technology, 68(10)105