

ActivityTracker

2020



FEBRUARY 7

COVENTRY UNIVERSITY

Authored by: VENKATA DEEPAK CHUDIPOTHU (9934621)



Logo
Name

Contents

Introduction.....	2
Software design & specification.....	2
Source Code.....	4
Software development approach.....	4
Development and deployment Challenges.....	4
Emerging Technologies	4
Screenshots	5

Introduction

ActivityTracker is a hybrid mobile app which is available on both Android and iOS platforms. It allows the users to add their activities and track them within the application. Users can set the activity status to completed as soon as they finish them. This application makes time management easier and effective.

Software design & specification

Requirements

User Story 1 –

As a user, I should be able to input my id and password and login to the system.

User Story 2 –

As a user, I should be able to see the “Add Activity” screen after login to the system.

The Add activity screen should have input fields for Activity name and description.

User Story 3 –

As a user, I should be able to save the activity details into the application.

User Story 4 –

As a user, I should be able to see all activities saved within the application as a list.

User Story 5 –

As a user, I should be able to set any particular activity as completed whenever it is done.

Components –

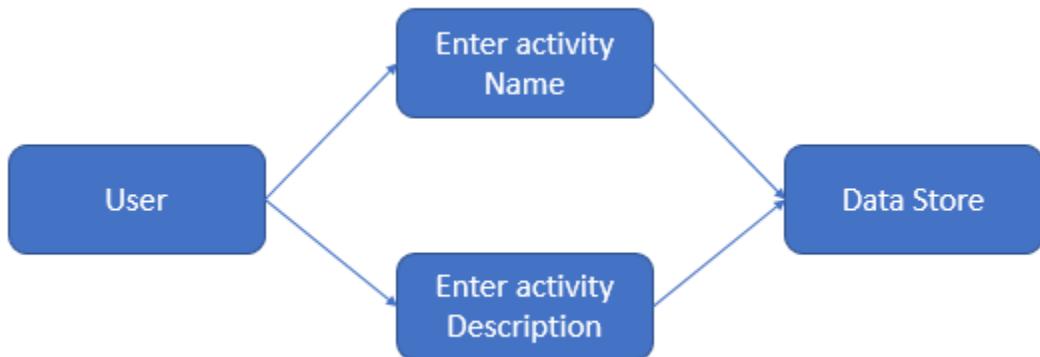
There are 3 major components of this project:

1. Activity files – These are the .xaml.cs files which contain the logic of the application. All event handlers and helper methods are present in these classes.
2. xaml files – These are the files which contain the code for the UI of the application.

3. Repository files – These are the files which contain methods to communicate with the database.

Data flow diagram

There is an input screen in the application and the data in the screen flows as shown in the diagram below



Project Setup Screenshots

A screenshot of Microsoft Visual Studio showing the development environment for the ActivityTracker project. The main window displays the XAML code for 'LogonPage.xaml' in the 'ContentPage' content area:

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:d="http://xamarin.com/schemas/2014/forms/design"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d"
    x:Class="ActivityTracker.LogonPage">
    <ContentPage.Content>
        <StackLayout Padding="20,20,20,20">
            <StackLayout Orientation="Horizontal">
                <Label Text="Username"/>
                <Entry x:Name="username" HorizontalOptions="Fill" Placeholder="Enter the username"/>
            </StackLayout>
            <StackLayout Orientation="Horizontal">
                <Label Text="Password"/>
                <Entry x:Name="password" HorizontalOptions="Fill" Placeholder="Enter the password" IsPassword="True"/>
            </StackLayout>
            <Button Text="Log On" Clicked="OnLogOnClicked"/>
            <Label x:Name="message"/>
        </StackLayout>
    </ContentPage.Content>
</ContentPage>
```

The 'Solution Explorer' pane on the right shows the project structure:

- Solution 'ActivityTracker' (1 project)
 - ActivityTracker
 - Dependencies
 - Models
 - Pages
 - ActivityRepository.cs
 - App.xaml
 - AssemblyInfo.cs
 - FileAccessHelper.cs
 - LogonPage.xaml
 - MainPage.xaml
- ActivityTracker.Android (incompatible)
- ActivityTracker.iOS (incompatible)

Project Team Configuration

We divided tasks among our team members as follows:

1. Requirements gathering, documentation and testing.
2. UI Development (Front-end development)
3. Database and backend development

Source Code

Source code is available on GitHub at below URL
<https://github.com/7052cemcw/ActivityTracker>

Software development approach

We have followed Agile mode of Software development. We started one user story at a time, we gathered the requirements of the user story, created the UI, implemented the logic and moved it for testing. We fixed the issues in that user story and moved forward.

Development and deployment Challenges

- **UI Alignment issues**

One major challenge that we faced while working on this project is that we found it a bit difficult to arrange the items in a proper layout on a screen.

- **Build issues**

If there is any error in .xaml file, it was difficult to identify the issue and build fails. C# errors are reported correctly by .NET platform, but .xaml errors could not be caught.

Emerging Technologies

Over the past years, the usage of mobile apps has increased, with this, more and more consumers are using apps for various purposes. Many technologies have emerged recently which allows the developers to create hybrid apps and reduces the burden to code an app multiple time. These technologies primarily use HTML5 to design the front end. Working in Hybrid technologies makes it easier for developers as they have to write the code once and run on all major platforms without extra efforts. Some of the emerging hybrid platforms are:

1. Xamarin – Founded in 2011, Microsoft owned company that uses .NET framework and C# as primary language to create the apps, it uses xaml to create the User interface of the applications. We have used Xamarin to create our project.

2. PhoneGap – It is the most popular cross-platform framework for mobile application development. It is open source framework backed by Adobe. It uses HTML, CSS and JavaScript to build the applications.

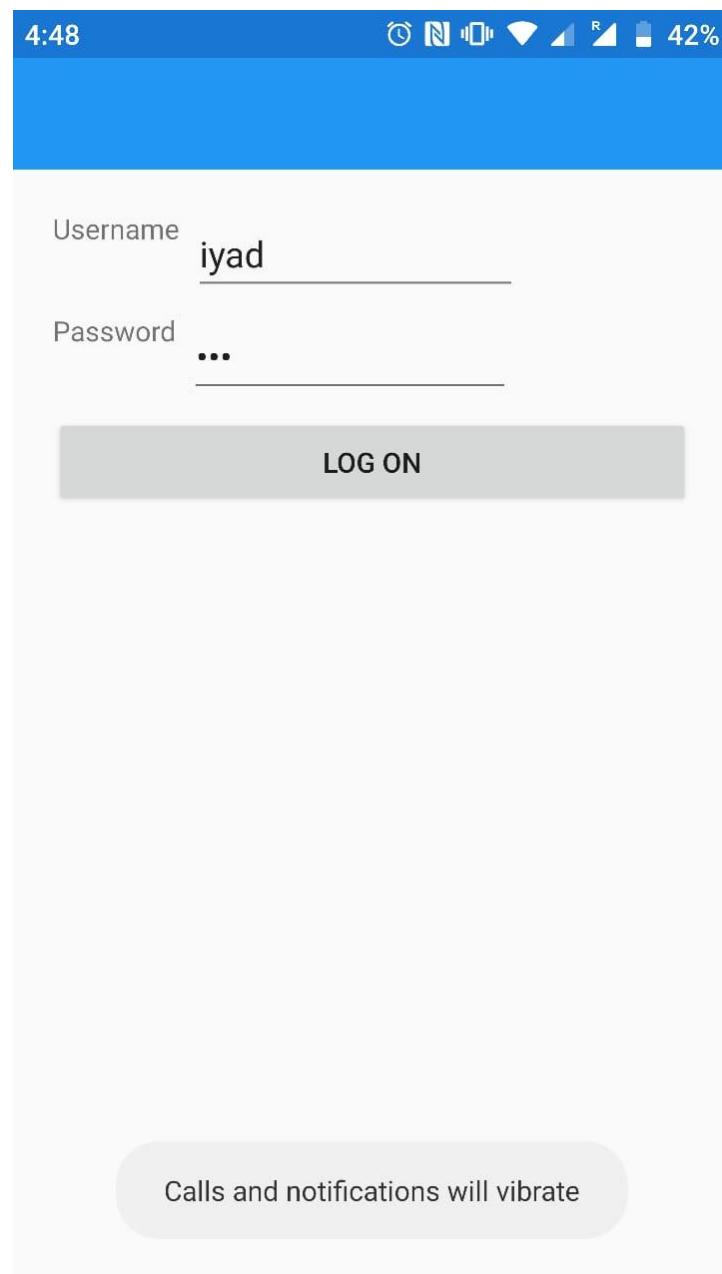
3. Ionic – Ionic is a free project licensed under MIT. It uses CSS, HTML5 and Sass to build the mobile application. Ionic is built using Angular.js and Apache Cordova.

4. React Native – It is a platform that allows the developers to build the applications by only using JavaScript. Using this framework, we can build an app which is indistinguishable from apps built using iOS or Android.

5. Flutter – It is Google's mobile framework which allows to craft high-quality native interfaces on Android and iOS in less time. It is free and open source.

Screenshots

Login Screen



Dashboard

