ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

MOBILE APP FOR COMMON HOME UTILITY SERVICE SYSTEM

P. Vishnu Mahesh [1], M. Harish Kumar [2], CH. Rani [3], L. Ashok Kumar [4], R. Harish [5].
1 Professor, Dept. of CSE, NS Raju Institute of Technology Visakhapatnam, Andhra Pradesh, India
2,3,4,5 Student, Dept of CSE, NS Raju Institute of Technology Visakhapatnam, Andhra Pradesh, India
vishnumaheshpenumerthy.cse@gmail.com, harishongala123@gmail.com, ashoklanka26@gmail.com, ranich0102@gmail.com, harishhd12345@gmail.com.

Abstract
Today, the life becomes busy and everyone wants their works to be done in no time. Besides livelihood most of the households frequently have to fix their house utility problems. Include electrical, plumbing, carpentry; Kitchen related utilities, Entertainment gadget etc., and other related problems. Many a number of places and times, though the quality of technicians available to fix or solve these problems, communication and information availability is becoming a hindrance.

Traditionally, people depend on personal connectivity, thru references and contacts to find the skilled technicians to get their problems solved. Time was not a barrier and people could wait a day or two to get problems fixed. But today time matters much and household utility problems cannot have leisure time. Today's modern gadgets need specialized and skilled technicians. Even though the society has lots of skilled manpower the lack of communication and information is distancing the people that need these services and the people that can provide these services. (Service Requester's Vs Service Provider's). This is a Society problem.

Today the technology and innovations are leading the business and in turn the world. With the availability of computing power, communication modes, hand held devices on one hand and the people's acceptance of these innovationson the other side is a good lead to solve the above problem. Today, almost every family in the society is having mobile phone connectivity and in some cases more than one phone per family. Wi-fi is ubiquity. So, there is need for availability of software that provides information regarding the available skilled manpower and technicians, their schedules, availability, prices, and specific expertise of technicians. This helps the households in choosing the right technician for their services in less time, with more efficient communicative methods and in turn may be with less cost. This software works like platform for bringing together Service Requester's Vs Service Provider's. CHUSS is an attempt to provide a "Mobile Application" that bridges the gap between Service Requester's Vs Service Provider's.

Cite this article as: P.Vishnu Mahesh, M.Harish Kumar, CH.Rani, L.Ashok Kumar & R.Harish, "Mobile App for Common Home Utility Service System", International Journal of Research in Advanced Computer Science Engineering, (IJRACSE), Volume 6 Issue 12, May 2021, Page 22-34.

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

CHUSS is an integrated system that provides necessary information to householders and skilled technicians. The goal and success parameter of CHUSS is the availability and capturing of the data relating to skilled technicians and also on the quality of the services CHUSS provides.

Keywords: Mobile App, Common Home Utility, Wi-fi is ubiquity.

1. Introduction

Today, electronic equipment (Home Utilities) is indispensable to our daily life. Electronic products revolutionized the lifestyle and gained a new level of popularity. From Ceiling Fan to Air Conditioner, we rely too much on them. The maintenance and fix these home utilities consumes lot of time. But today time matters much and household utility problems cannot have leisure time. The Home utility problems include electrical, plumbing, carpentry; Kitchen related utilities, Entertainment gadget etc., and other related problems. To overcome this, we can take the advantage of technology, develop and apply better software applications. This prompted us to think for a better and portable Application known as "Common Home Utility Service System" (CHUSS).

1.1 Proposed System

Today the technology and innovations are leading the business and in turn the world. With the availability of computing power, communication modes, hand held devices on one hand and the people's acceptance of these innovations on the other side is a good lead to solve the above problem. Today, almost every family in the society is having mobile phone connectivity and, in some cases, more than one phone per family. Wi-fi is ubiquity. So, there is need for availability of software that provides information regarding the available skilled manpower and technicians, their schedules, availability, prices, and specific expertise of technicians. This helps the households in choosing the right technician for their services in less time, with more efficient communicative methods and in turn may be with less cost. This software works like platform for bringing together Service Requester's Vs Service Provider's.

CHUSS is an attempt to provide a "Mobile Application" that bridges the gap between Service Requester's Vs Service Provider's. CHUSS is an integrated system that provides necessary information to householders and skilled technicians. The goal and success parameter of CHUSS is the availability and capturing of the data relating to skilled technicians and also on the quality of the services CHUSS provides.

The scope of this project work is to develop a prototype of application with basic Modules. The main Functionality of Modules, data Base, basic user queries are implemented. Though in reality we need live data, we tested the application with our own test data. Due to time and other resource constrains, this is only a proof of Concept. Traditionally, people had to make calls or need to find a technician in person, then wait for his availability. Sometimes the technician doesn't show up too. The best solution is switching over to CHUSS (Find My Technician) based on the wide range of

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

utility catalog offered by CHUSS. CHUSS works with the local skilled technicians, and that means significant money savings yet with a professional service. In a nutshell, the customers choose convenience at their fingertips. Virtually anyone with a smart phone can book for a technician. It's just one click away. A reasonable and Non-Negotiable price makes the "pricing" hassle free for the customers. Convenient payment mode Pay On delivery (POD) makes CHUSS the transparent in pricing. Opportunity to earn depending on the work done. Freelance while you run other sort of businesses. Offers time based scheduled work, which makes it feasible for many.

1.2 Feasibility Study

The statements "don't try to fix it unless you understand it" apply describe the feasibility study stage of system analysis. The goal of feasibility study is to evaluate alternative system and to purpose the most feasible and desirable system for development.

It consists of the following:

1. Statement of the problem
2. Summary of findings and recommendations
3. Details of findings
4. Recommendations and conclusions

There are basically five types of feasibility are addressed in the study.

1. Technical feasibility
2. Economic feasibility
3. Motivational feasibility
4. Schedule feasibility
5. Operational feasibility

1.3 Technical Feasibility:

It is technically feasible to design the project as; the entire modules described in the modules description can be created using Front-End and backend using DART.

Advantages of DART

- Same Dart Scripts Work in iOS and Android Native Apps Without Modification. ...
- Compile to Self-contained Snapshot. ...
- Static Typing, Finding Errors, and Optimization. ...
- Null Safety and Reducing App Crashes. ...
- Dart is Supported by Google.

Economic Feasibility

- In our project to find the cost of our project the following things are needed
- Number of the persons(N)=3
- Effort in person cost(E)=2000
- Time period(t)=3 moths
- Interface of the project(I)=10% of E
- Number of modules(M)=2
- Software cost(SC)=10000*5%(including OS, Front end,Back end)
- Project Cost=((N*E)+(I*M)+SC)*t=((3*2000) +((2000*0.1)*2)+(10,000*0.05)*3.

2. System Analysis

2.1 Introduction

Term system is derived from the Greek word 'Systema' which means an organized relationship among functioning units or components. A system is an orderly grouping of interdependent components linked

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

together according to a plan to achieve a specific objective.

Characteristics of a System

a) Organization
b) Interaction
c) Interdependence
d) Integration
e) CentralObjective

Organization-It implies structure and order.

Interaction- It refers to manner in which each component functions with other components of the system.

Interdependence- Units / parts are dependent on each other.

Integration The parts of a system work together within the system even though each part performs a unique function.

Central Objective Objective may be real or stated. All the components work together to achieve that particular objective.

Element sofa system

- Output sand Inputs
- Processor
- Control
- Feedback
- Environment
- Boundaries and Interface

Inputs and Outputs-Inputs are the elements that enter the system for processing and output is the result of processing.

Processor- It is the element that involves the actual transformation of input into output.

Control- The control element guides the system.

Feedback- Output is compared against performance standards.

Environment It is the supra system within which an organization operates.

Boundaries and Interface-A system should be defined by its limits.

3. Software Development Life Cycle

Information system development involves various activities performed together. The stages involved during System Life Cycle are:

a) Planning (Recognition of need , Feasibility study
b) Analysis
c) Design
d) Implementation
e) Testing & Integration
f) Post implementation and maintenance

Planning (Recognition of need):

It is the first stage of information system development cycle. This gives a clearer picture of what actually the existing system is the preliminary investigation must define the scope of the project and the perceived problems, opportunities and directives that triggered the project.

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

The preliminary investigation includes the following tasks:

- List problems, opportunities and directives.
- Negotiate preliminary scope.
- Assess project worth.
- Plan the project.
- Present the project and plan.

System Analysis

System analysis is not a preliminary study. It is an in-depth study of end user information needs that produces functional requirements that are used as the basis for the design of a new information system. System involves detailed study of:

a. The information needs of the Organization /Application and End User.
b. The activities, resources and products of any present information system.
c. The information system capabilities required to meet our Information needs.

The Other Three Phases are explain their respective Chapters.

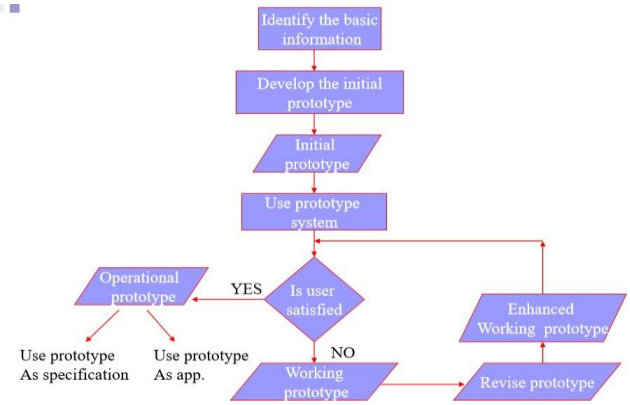
SDLC Methodologies and Approaches:

There are several SDLC Methodologies that can be applied to various software applications. They include:

1. Waterfall Model
2. Object Oriented Model
3. Iterative Incremental Model
4. Spiral Model
5. Prototyping Model
6. Agile Development Model
7. DevOps Developmen tModel

All these Development model have their own advantages and disadvantages. We have selected "Prototyping Model" for our IGMHS Project.

Prototyping



Prototyping is the process of building a model system. In terms of an information system prototype are employed to help system designers build an information system that intuitive & easy to manipulate end users. Prototyping is an iterative process that is part of analysis phase of system development of life cycle.

Advantages of prototyping

- Prototype reduce the development time.
- It reduce the development cost.
- It requires the users involvement.
- Its results are higher user satisfaction.
- Prototype are an active, not passive, model that end user can see, touch & feel.
- Prototyping can increase creativity because it allows for quicker user feedback which can lead to better solutions.

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

3.1 System requirements for chuss

These are the requirements we developed by discussions and brainstorming among the project team members. We took some inputs from our other project colleagues, College staff and project guide.

3.2 Functional requirements:

In software engineering and systems engineering, a functional requirement defines a function of a system or its component, where a function is described as a specification of behaviour between outputs and inputs.

1. Integrating Global Medical Data or Data Capturing.
2. Integrating Heterogeneous Data.
3. Data Cleaning or Data Pre-processing.
4. Queries and Reporting.
5. Test Data Preparation and Testing Applications.

3.3 Non-Functional requirements:

In systems engineering and requirements engineering, anon-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions.

1. Login and Authentication.
2. Security.
3. Backup and Recovery.

3.4 Software requirements

1. Operating System : Android
2. Database : Google Firebase
3. Languages : DART

3.5 Hard ware requirements

The minimum hardware requirement specifications for developing this project are:

- Processor : Standard processor with speedof3.0GHz
- RAM : 4 GB RAM or higher
- Free Storage : 4GBormore (SSD is strongly recommended)
- Monitor : Standard color monitor
- Screen Resolution : 1920 x 1080
- Keyboard : Standard keyboard
- Mouse : Standard mouse

3.5 Dart

Dart is a client-optimized language for developing fast apps on any platform. Its goal is to offer the most productive programming language for multi-platform development, paired with a flexible execution runtime platform for app frameworks.

Dart is an open-source programming language.

Dart is cross-platform and supports all major operating systems.

The Dart code can be trans-compiled into JavaScript using source-to-source compilation.

The Dart is type safe programming language.

Dart supports Ahead-Of-Time (AOT) compilation mode.

Dart cade can be compiled optimized

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

JavaScript code that can be executed on all modern web browsers.

Dart is unusually flexible at compilation and execution.

Dart supports asynchronous programming which let your program run without getting blocked.

Dart is an asynchronous programming language.

Development

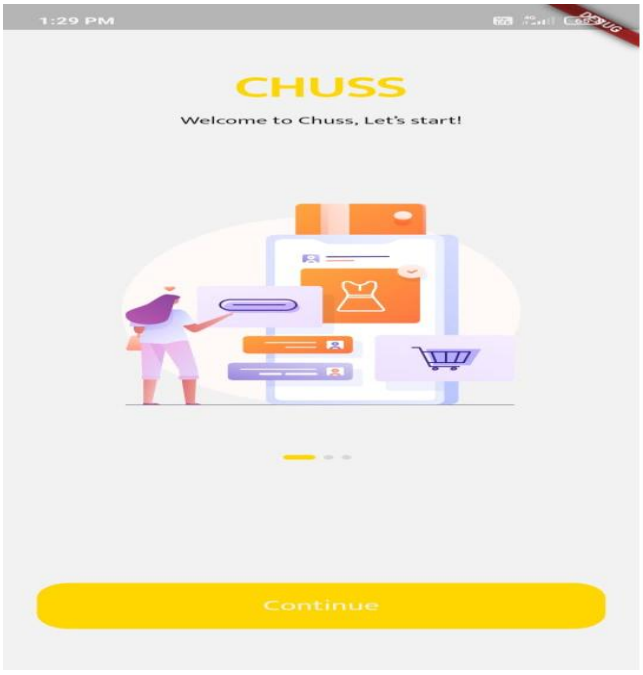
4. Introduction

Software development is the process of conceiving, specifying, designing, programming, documenting, testing, and bug fixing involved in creating and maintaining applications, frameworks, or other software components.

4.1 CHUSS –Input Screen

4.2 Output Screen

S1: About CHUSS



Execution for the sample code

Sample Code contains for the **CHUSS**

Sample code: TechHome.dart

```
import 'package:chuss/components/loginmain.
    dart';
import 'package:firebase_auth/firebas
    e_auth.dart';
import 'package:flutter/material.dart';

class TechHome extends StatefulWidget {
 const TechHome({Key key}) : super(key: ke
    y);



 @override
 _TechHomeState createState() => _TechHo
    meState();
}

class _TechHomeState extends State<TechHo
    me> {

 FirebaseAuth auth = FirebaseAuth.instance;

 signOut() async {
  await auth.signOut();
 }


 @override
 Widget build(BuildContext context) {
  return Scaffold(
   body: SingleChildScrollView(
    child: Center(
     child: ElevatedButton(onPressed: () {
      signOut();
      Navigator.of(context).pushAndRemov
   eUntil(
```

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

```
            MaterialPageRoute(
              builder: (BuildContext context) =>
    LoginMain(),
            ),
              (route) => false);

        }, child: Text('logout'),),
      ),
    ),
   );
  }
}
```

Sample code: Body.dart

```
import 'package:chuss/components/constants.d
    art';
import 'package:chuss/components/default_bu
    tton.dart';
import 'package:chuss/components/loginmain.
    dart';
import 'package:flutter/material.dart';
import 'package:chuss/splashscreen/componen
    t/splash_content.dart';

import '../../size_config.dart';


class Body extends StatefulWidget {
 @override
 _BodyState createState() => _BodyState();
}

class _BodyState extends State<Body> {

 int currentPage = 0;
 List<Map<String, String>> splashData = [
  {
   "text": "Welcome to Chuss, Let's start!",
   "image": "assets/images/splash_1.png"
  },
  {
   "text":
       "We help people connect with Local T
   echnicians \naround them",
   "image": "assets/images/splash_2.png"
  },
  {
   "text": "We show the easy way to grab a te
   chnician. \nJust stay at home with us",
   "image": "assets/images/splash_3.png"
  },
 ];
```

System Testing

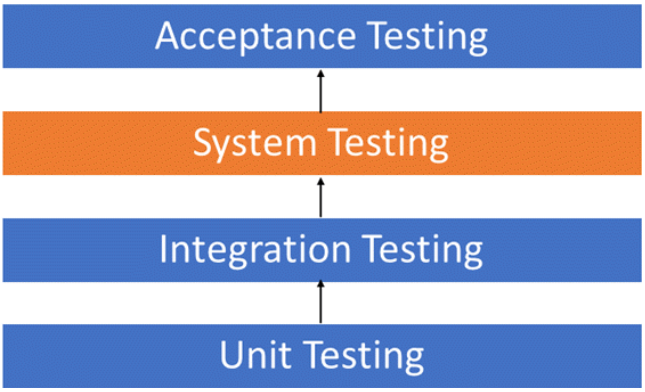
5.1 Introduction:

System Testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system. Ultimately, the software is inter faced with other software/hardware systems. System Testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system. There are different types of Tests performed on Software Code, some of the important type softesting's and their hierarchy are:

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

5.2 Types of Testing:

Software Testing Hierarchy:



As with almost any software engineering process, software testing has a prescribed order in which things should be done. The following is a list of software testing categories arranged in chronological order. These are the steps taken to fully test new software in preparation for marketing it:

- Unit testing performed on each module or block of code during development. Unit Testing is normally done by the programmer who writes the code.
- Integration testing done before, during and after integration of a new module into the main software package. This involves testing of each individual code module. One piece of software can contain several modules which are often created by several different programmers. It is crucial to test each module's effecton then tire program model.
- System testing done by a professional testing agent on the completed software product before it is introduced to the market.
- Acceptance testing-beta testing of the product done by the actual end users.

5.2.1 Unit testing:

UNIT TESTING is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. Inprocedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouragedas there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

Unit Testing Method:

- It is performed by using the White Box Testing method.
- Unit testing is the first level of software testing and is performed prior to Integration Testing.
- It is normally performed by software developers themselves or their peers. In rare cases, it may also be performed by in dependent software testers.

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

Integration Testing is defined as a type of testing where software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated. Integration Testing focuses on checking data communication amongst these modules. Hence it is also termed as 'I&T'(Integration and Testing),'String Testing' and sometimes 'Thread Testing'.

6. Implementation

Implementation simply means carrying out the activities described in your workplan. Executing a project in the water and sanitation sector is a very complex mission, as it requires the coordination of a wide range of activities, the overseeing of a team, the management of budget, the communication to the public, among other issues. Independent of whether it is asocial project to raise the awareness and promote hygiene or it is a construction project for service delivery, there is a certain process that has to be followed. The following lines will give you an introduction into the implementation of projectsin sustainable sanitation and water management, and highlights key aspects that have to be taken into account for a successfully implementation.

Advantages:

Implementation gives the opportunity to see the plans become are ablity Execution of projects allows end-users to have access to better services and living environment Success stories and experiences can be shared with specialists from other cities and towns, encour aging others to adopt similar approaches, which in turn may improve water resources management in the local area

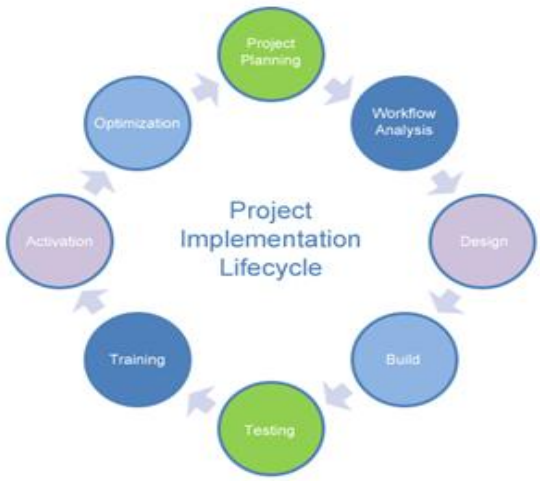
Objective soft he Implementation Phase:

Factsheet Block Body

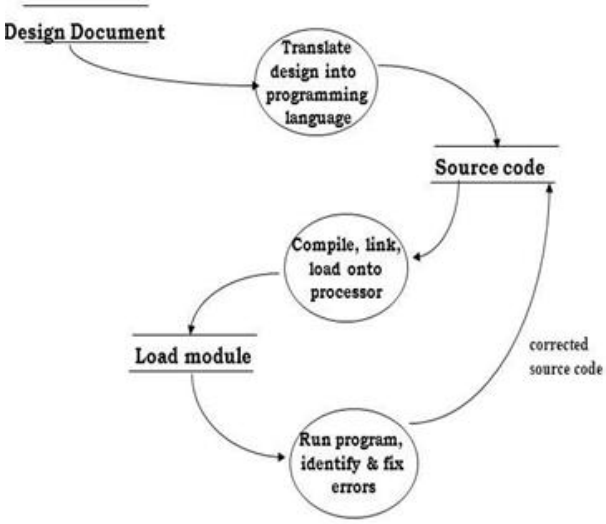
The objectives of the implementation phase can be summarized as follow:

- Putting the **action plan** into operation.
- Achieving tangible change and improvements.
- Ensuring that new infra structure, new institution sand new resources are sustainable in every aspect.
- Ensuring that any unforeseen **conflicts** that might arise during this stage are resolved.
- Ensuring transparency with regard to **finances**.

Ensuring that potential benefits are not captured by elites at the expenses of poorer social groups.



ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com



Technologies

Overview of Technologies

Software Environment is a technical specification of requirement of software product. This specifies the environment for development, operation and maintenance of the product.

Technology used:

- Flutter
- Dart
- Google Firebase

7. Overview of Flutter:

Flutter is a cross-platform UI toolkit that is designed to allow code reuse across operating systems such as iOS and Android, while also allowing applications to interface directly with underlying platform services. The goal is to enable developers to deliver high-performance apps that feel natural on different platforms, embracing differences where they exist while sharing as much code as possible.

During development, Flutter apps run in a VM that offers stateful hot reload of changes without needing a full recompile. For release, Flutter apps are compiled directly to machine code, whether Intel x64 or ARM instructions, or to JavaScript if targeting the web. The framework is open source, with a permissive BSD license, and has a thriving ecosystem of third-party packages that supplement the core library functionality.

This overview is divided into a number of sections:

1. The **layer model**: The pieces from which Flutter is constructed.
2. **Reactive user interfaces**: A core concept for Flutter user interface development.
3. An introduction to **widgets**: The fundamental building blocks of Flutter user interfaces.
4. The **rendering process**: How Flutter turns UI code into pixels.
5. An overview of the **platform embedders**: The code that lets mobile and desktop OSes execute Flutter apps.
6. **Integrating Flutter with other code**: Information about different techniques available to Flutter apps.
7. **Support for the web**: Concluding remarks about the characteristics of Flutter in a browser environment.

DART

Dart is a client-optimized language for developing fast apps on any platform. Its goal is to offer the most productive programming language for multi-platform development, paired with a flexible execution runtime platform for app frameworks. Languages are defined by their *technical envelope* — the

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

choices made during development that shape the capabilities and strengths of a language. Dart is designed for a technical envelope that is particularly suited to client development, prioritizing both development (sub-second stateful hot reload) and high-quality production experiences across a wide variety of compilation targets (web, mobile, and desktop). Dart also forms the foundation of Flutter. Dart provides the language and runtimes that power Flutter apps, but Dart also supports many core developer tasks like formatting, analysing, and testing code.

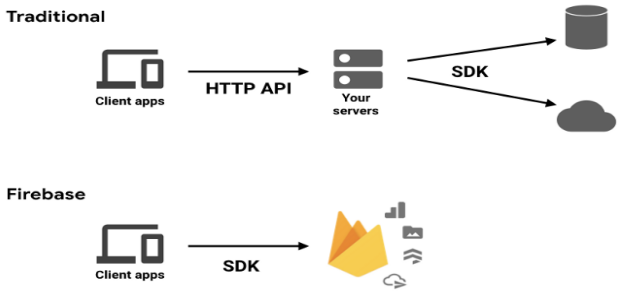
Dart: The language

The Dart language is type safe; it uses static type checking to ensure that a variable's value *always* matches the variable's static type. Sometimes, this is referred to as sound typing. Although types are mandatory, type annotations are optional because of type inference. The Dart typing system is also flexible, allowing the use of a dynamic type combined with runtime checks, which can be useful during experimentation or for code that needs to be especially dynamic.

Dart offers sound null safety, meaning that values can't be null unless you say they can be. With sound null safety, Dart can protect you from null exceptions at runtime through static code analysis. Unlike many other null-safe languages, when Dart determines that a variable is non-nullable, that variable is *always* non-nullable. If you inspect your running code in the debugger, you'll see that non-nullability is retained at runtime (hence *sound* null safety).

Google Firebase

Firebase is a toolset to "build, improve, and grow your app", and the tools it gives you cover a large portion of the services that developers would normally have to build themselves, but don't really want to build, because they'd rather be focusing on the app experience itself. This includes things like analytics, authentication, databases, configuration, file storage, push messaging, and the list goes on. The services are hosted in the cloud, and scale with little to no effort on the part of the developer. This is different than traditional app development, which typically involves writing *both* frontend and backend software. The frontend code just invokes API endpoints exposed by the backend, and the backend code actually does the work. However, with Firebase products, the traditional backend is bypassed, putting the work into the client. Administrative access to each of these products is provided by the Firebase console.



8. Conclusion:

As a prototype project we tried, implementing all features required for a full-fledged software project. The Initial Planning, Analysis and Design phases gave us good analytical skills. In problem

ISSN No : 2454-4221 (Print)
ISSN No : 2454-423X (Online)
International Journal of Research in Advanced Computer Science Engineering
A Peer Reviewed Open Access International Journal
www.ijracse.com

identification phase we had good brainstorming capabilities and increased our thinking capabilities. In design phase, we learned the practical utility of software engineering concepts and their applications in live projects. The development stage was interesting as we have to learn new concepts like connectivity of client code with data base server, and parameterize middleware with appropriate port numbers, host names, database name etc., The testing was good as we could see the output and in good number of case exceptions and fixing logic of the code. Though we did not have full experience of implementing live, but porting the application, setting the environment in new hardware was interesting. The implementation taught us the use fullness of scripting some additional code like table creation and windows batch files.

In totally, the involving and doing software project was good learning, interesting and inspiring. Though the reality of chuss is dreamy, huge task, lots of barriers but still we are positive and think there must be a start for everything.

Performance:

CHUSS is developed in Flutter technology is a versatile and platform independent. The features provided by the CHUSS makes easy to interact with technicians.

Enhancements:

Technician availability, as CHUSS works nearby location-based technicians. Technicians panel can be more enhanced.

References:

Software Engineering:

1. Software Engineering: APrimer,WamanSJawadekar,TataMcGraw-Hill,2008
2. Software Engineering, a Precise Approach, PankajJalote, WileyIndia, 2010

DART:

1. Dart Apprentice, By Jonathan Sande & Matt Galloway.
2. Scalable Application Development, By Davy Mitchell, Sergey Akopkokhyants, Ivo Balbaert.
3. Dart in Action, By Chris Buckett.

Sftware Project Management:

1. Bob Hughes & Mike Cotterell, TATA Mcgraw-Hill
2. Software Project Management, Walker Royce: PearsonEducation,2005.