**4**

Software and Applications

In this chapter we will discuss about various phases of programming language used for development of this project. To understand the process we go through several topics which best describe about the project. Firstly, we will discuss about Software Development Process or Software Development Life Cycle used. Secondly, we go through which programming languages used in this project and why? Then we discuss the VCS(Version Control System) used in this project while project code implementation, development and for maintenance.

In this project we used LAMP model for hosting our web service which can be accessed via android app. As told LAMP model is used, we created our project on *Beaglebone Black* which is Linux based machine and we hosted our service with the help of Apache server which makes php scripts able to be executed with the help of URI and have MySQL which holds all the scheduled jobs data generated by client. LAMP is widely used and known for *open source solution stack* for *web based services* as its all four components are open-source.

**4.1. Software Development Life Cycle (SDLC)**

**4.1.1. INTRODUCTION**

The systems development life cycle (**SDLC**) is a conceptual model used in project management that describes the stages involved in an information system development project, from an initial feasibility study through maintenance of the completed application.

There are various software development life cycle models defined and designed which are followed during software development process. These models are also referred as "Software Development Process Models". Each process model follows a Series of steps unique to its type, in order to ensure success in process of software development.

**4.1.2. Types of SDLC Models**

Following are the most important and popular SDLC models followed in the industry:

* Waterfall Model
* Iterative Model
* Spiral Model
* V-Model
* Big Bang Model

The other related methodologies are Agile Model, RAD Model, Rapid Application Development and Prototyping Models.

**4.1.3. SDLC Model Chosen**

While taking some aspects of this project length and team members in this project it is best suited to choose Big Bang Model for this project software code development.

The Big Bang model is SDLC model where there is no specific process followed. The development just starts with the required money and efforts as the input, and the output is the software developed which may or may not be as per customer requirement.

There is no formal development followed and very little planning is required. Even the customer is not sure about what exactly he wants and the requirements are implemented on the fly without much analysis.

Usually this model is followed for projects where the development teams are very small. While other model don’t support the **nonlinear** software development process.

**4.1.4. Why not other models**

The main reason of not selecting other model because they don’t support nonlinear software development process for a small team.

Big Bang model provides out team to develop software in nonlinear manner because we are getting our resources in random order and we need to start development process.

**4.2. Introduction to Programming Languages & Structured Query Languages Used**

In this topic we describe shortly about structured queried language for handling users data in form of database and programming languages used as back-end and front-end in our project.

**4.2.1. Structure Query Language (SQL)**

SQL is a special-purpose programming language designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS).

For our project we used MySQL as a DBMS and further details are given in *section 4.5.1.* .

**4.2.2. Backend Programming Languages**

For backend data processing we used two scripting languages Shell Scripts and PHP Scripts which control or handle all the data generated via end user for controlling appliances.

4.2.2.1. Shell Scripts

A shell script is a computer program designed to be run by the UNIX shell, a command-line interpreter. The various dialects of shell scripts are considered to be scripting languages.

In our project some files are required to be a member of www-data and should have permission to read and write files, so we created some startup scripts which executes at boot up time setup the appropriate environment for end user.

4.2.2.2. PHP Scripts

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. PHP originally stood for Personal Home Page, but it now stands for the recursive backronym PHP: Hypertext Preprocessor.

In our project server or we can say gateway is powered by php, it is powered by apache servers by which php scripts can be accessed as a web content. Which in turn access GPIO Api of Beaglebone Black to control appliances.

g

**4.2.3. Front end programming language**

To create front end software we used android devices to launch our app by which user can easily get access to their appliances.

Android software development is the process by which new applications are created for the Android operating system. Applications are usually developed in Java programming language using the Android software development kit (SDK), but other development environments are also available.

From this point we will consider our front end programming language as **Android App Programming**.

**4.3. Software Utilities**

Some software utilities are used in our project to reduce the complexity of our project development. We used both pre-installed and third party utility of LINUX OS.

Software utilities used:

* CRON
* PHP-JOB-SCHEDULER

As we know both are used for same purpose but in our project used both **experimentally** to reduce the complexity of CRON in our project which help of PHP-JOB-SCHEDULER and used CRON to overcome the limitation of PHP-JOB-SCHEDULER.

**4.4. Introduction to Utility Software**

Utility software, along with operating system software, is a type of system software used to support the computer infrastructure, distinguishing it from application software which is aimed at directly performing tasks that benefit ordinary users.

We employed both OS utility (CRON) and third-party utility software (PHP-JOB-SCHEDULER) for our project.

g

**4.4.1. Crontab Utility**

In this topic a brief description is given about cron jobs or crontab. Why it is employed? Limitation of it and how to overcome form those limitations? And how they are used in our project.

4.4.1.1 Introduction

The software utility Cron is a time-based job scheduler in Unix-like computer operating systems. People who set up and maintain software environments use cron to schedule jobs (commands or shell scripts) to run periodically at fixed times, dates, or intervals. It typically automates system maintenance or administration—though its general-purpose nature makes it useful for things like downloading files from the Internet and downloading email at regular intervals. The origin of the name cron is from the Greek word for time (chronos).

4.4.1.2 Why Cron employed?

Cron is used in our project because of two main reasons:

* Execute a script named *“export.sh”*.This file set up an environment for user to get access to read and write some files using web request.
* Execute a php script every 30 seconds named *“firepjs.php”.* This file checks for the available schedules of tasks in database and execute them at same time.

Every 10 second to execute a file used only for development purpose it can be set to 1 minute because users are only allowed to define time in at least minutes only.

4.4.1.3 Advantages of Cron

1. The main advantage of crontab jobs in this project is that it provides a facility to execute a script only once when system is started.

2. Another advantage of crontab jobs is that they are controlled by OS hence no human interface is further required for jobs execution.

4.4.1.4. Limitations of Cron

1. The main limitation or disadvantage of it’s working in this project is its complexity of program to manage all the jobs.

a. A separate complex program needed to be written to handle addition of process.

b. After execution a separate program is needed to employ to remove job entry from the crontab.

2. Program which keep tract of jobs of user need to be added in crontab which maintain the list by which it can decide to repeat or pause or resume any scheduled job. Due this complexity of program increased.

To overcome from all these issues we employed a less complex job scheduler called PHPJOBSCHEDULER.

4.4.1.5. Crontab jobs list

As detailed in 4.4.1.2 there are two jobs need to be scheduled one at boot up time and other at every 30 seconds.

As we know that it is not possible to give a command to cron jobs to be executed in seconds. The least unit is in minutes. So, we modified the second job command al little bit as shown below:

1. @reboot /var/www/php\_project/export.sh
2. \* \* \* \* \* php5 /var/www/phpjobscheduler/firepjs.php; sleep 30; php5 /var/www/phpjobscheduler/firepjs.php

These two entries are need to me made in job file and can be written with the help of

root ~ $ crontab –e

After typing command *crontab –e* press enter and type in both the statements shown above.

Statement 1 have *@reboot* in starting it means whenever system reboots or boots up it execute the defined *export.sh* shell script only once per boot.

Statement 2 is executes every 1 minute but inter sleep makes it execute firepjs.php in starting of every minute and delay for 30 second and then at half of a minute execute the same script again.

g

**4.4.2. PHP Job Scheduler Utility**

In this topic a brief description is given about cron jobs or crontab. Why it is employed? Limitation of it and how to overcome form those limitations? And how implemented in our project.

4.4.2.1. Introduction

Design to automate tasks by scheduling PHP scripts to run at set intervals. It is a replacement for cron jobs on UNIX **or** scheduled tasks using Microsoft Scheduler. By using it we can schedule for minutes, hours, days, weeks and so on and can handle repetition like pause and play repetition.

It is open source web based scheduler which requires PHP and MySQL services to work and it can execute PHP scripts which in turn perform tasks which we define in those scripts. It makes logs of those executions in database and can be seen later for debugging purpose. It is easy to set up and have less complexity to create a scheduled task in comparison with CRON jobs. It makes easy to handle all scheduled task separately without affecting other scheduled task.

4.4.2.2 Why PHPJOBSCHEDULER employed?

PHP job scheduler is used in our project as a scheduler for appliances or any switch in home.

* Constantly checks for any available task available in database.
* File which turn on or off switch is “*task.php”* which have control to turn on or off the appliances.

4.4.2.3 Advantages of PHP Job Scheduler

1. The main advantage of this is that, it provides easy creation of schedule and managing them.
2. It provides a special feature that it makes any job schedule to be pause and can be resumed later as per user requirement.
3. No more further requirement of writing complex codes to create and manage cron job schedules.

4.4.1.4. Limitations of Cron

The only limitation of php job scheduler is that it never check itself for any job to be executed, user need to execute a php script *“firepjs.php”* to be executed manually or embedded in any existing scripts which is being generally executed by client or user periodically.

To overcome from this limitation the only solution left which means *“firepjs.php”* is to be scheduled with *CRON* and hence this is required only once to make a single schedule in it which executes *firepjs* script after 30Sec or 10Sec or 1Min as per the programmer need.

**4.5. Software Brief Description**

In this section all the things and aspect described in *section 4.2* would be consider in brief. Firstly we will discuss in detail about uses of MySQL database server in our project after which we will discuss about role of PHP, Shell scripts in our project which is used as back-end programming language and finally give detailed theory of android app used in our project as a front-end programming language.

We discuss previously described Structured Queried/Programming Languages in following sequence:-

* MySQL
* Back-End Programming languages
  + Shell Scripts
  + PHP Scripts
* Front-End Programming language
  + Android

g

**4.5.1. MySQL**

4.5.1.1. Introduction

MySQL is named after Michael Widenius (who is a co-founder of MySQL) daughter, My, while "SQL" stands as the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. For proprietary use, several paid editions are available, and offer additional functionality.

4.5.1.2. Use of MySQL in project

MySQL is used for storing the data into database of schedules generated by user for turning on and off the switches on particular time.

In this project a third party job scheduler utility is used which executes tasks on time which requires MySQL database to store and retrieve schedules entries.

**4.6. Project Source Code Management System**

In this project we used a source code management system by which we can keep track of our software code growth in better manner and can assess our team efficiency of software development.

We used GIT as a content tracker or Source Code Management System. It is a *distributed revision control system* with an emphasis on *speed*, *data integrity*, and *support for distributed*, *non-linear* workflows.

We used it to keep track of all our source code of back end and front end software’s.

**4.6.1. Why GIT?**

g

We used ‘git’ because of two reasons:

1. First one is that we need to keep track of all our progress of software development with collaborating with other team members and,
2. Second reason is that we used all the things from hardware to programming language or OS are open-source and this source content tracker must be open source, and of course due to its simplicity too.

g

**4.6.2. Structure of software code repository**

In this section we describe about the structure of our software code repository:

* php\_project Repository (Back-End Software)
* KuchBhi Repository (Front-End Software)

As we collaborate within our team we need a common place from where we can sync our project workflow with each other. For this we used a free and a private cloud service of [*https://bitbucket.com*](https://bitbucket.com) to collaborate. Our url is [*https://bitbucket.org/etechclub/*](https://bitbucket.org/etechclub/)where our all codes belonging to this project is placed.

4.6.2.1. php\_project Repository

* php\_project
  + .git […]
  + Scripts
* db
* connect.php
* newschedule.php
  + - check.php
* switch.php
* task.php
* dumey.php
* index.php
* .gitignore

4.6.2.1. KuchBhi Repository

* KuchBhi
  + .gradle […]
  + .idea […]
  + App
* build […]
* libs […]
* src
  + Main
    - .git […]
    - java
      * com\etechclub\kuchbhi
        + CompleteConnection.java
        + ControlPanel.java
        + Main2Activity.java
        + Main22Activity.java
        + MainActivity.java
        + SetupConnection.java
    - res
      * drawable […]
      * layout
        + activity\_main.xml
        + activity\_main2.xml
        + activity\_main22.xml
        + complete\_connection.xml
        + controlpanel.xml
      * minmap-hdpi […]
      * minmap-mdpi […]
      * minmap-xhdpi […]
      * values
        + colors.xml
        + dimens.xml
        + strings.xml
        + styles.xml
      * values-w820p […]
  + .gitignore
  + AndroidManifest.xml
    - * app.iml
      * build.gradle
      * progurard-rules.pro
  + build […]
  + gradle […]
* buil.gradle
* gradle.properties
* gradlew
* gradlew.bat
* KuchBhi.iml
* local.properties
* settings.gradle