Principles of Software Engineering
SE 302

Project Group 6

Project Document

Version 1.0

29 December 2015

NOTICE-BOARD WITH RASPBERRY PI

Project Team: Emre ÖVÜNÇ
Ayşe Simge ÖZGER
Buse ÖZER
Mert SOLAK
Onur ELGÜN

Project Manager & Customer: Kaya OĞUZ

Computer Engineering, Izmir University of Economics

CONTENTS

1 INTRODUCTION

This document contains the software requirements for the NoticeBoard with Raspberry Pi system. In this project user can announce his messages using internet. User can send messages from his phone or PC, and the message will show on LCD screen which is mounted at his/her office door.

1.1 Preface

Our purposes are easy access to students and announcement convenience for the user. This project will provide a user-friendly interface and fast data transmission between user's device and LCD screen via Raspberry Pi.

1.2 Roles & Responsibilities

Emre Övünç – TweetBot to Check Tweets

Ayşe Simge Özger – Raspberry Pi FTP Server Installing

Onur Elgün – Design Interface for Computers

Buse Özer – LCD Panel Connection

Mert Solak – Design Interface for Mobile Platforms

1.3 Scope

Raspbian OS is installed in the Raspberry Pi B+ which has SD card, power adapter and usm wireless modem. Raspberry Pi provides checking twitter account, display messages on Led Panel. Also, Led Panel connects to Raspberry Pi with cables. Raspberry Pi has an internet connections by using usb wireless modem. Raspberry Pi, smart phones application and computer application communicate each other with FreeDNS services. In addition, in the Raspberry Pi crontab is used for dns services and TweetBot program. User only need to internet connection to using program and sending messages.

2 GLOSSARY

Requirements analysis: The process of classifying requirements information into various categories, evaluating requirements for desirable qualities, representing requirements in different forms, deriving detailed requirements from high-level requirements, negotiating priorities, and so on.

Customer: A project stakeholder who requests, pays for, selects, specifies, uses, or receives the output generated by a product.

System requirement: A top-level requirement for a product that contains multiple subsystems, which could be all-software or software and hardware.

User requirement: User goals or tasks that users must be able to perform with a system, or statements of the user's expectations of system quality.

Data-flow diagram (DFD). A diagram that shows the movement of data between processes, entities, and data stores within a system.

3 OVERVİEW

3.1 Project Summary

These project's aim easy access to students and announcement convenience for the user. This project will provide a user-friendly interface and fast data transmission between user's device and LCD screen via Raspberry Pi.

Raspbian OS is installed in the Raspberry Pi B+ which has SD card, power adapter and usb wireless modem. Raspberry Pi provides checking twitter account, display messages on Led Panel. Also, Led Panel connects to Raspberry Pi with cables. Raspberry Pi has an internet connections by using usb wireless modem. Raspberry Pi, smart phones application and computer application communicate each other with FreeDNS services. In addition, in the Raspberry Pi crontab is used for dns services and TweetBot program. User only need to internet connection to using program and sending messages.

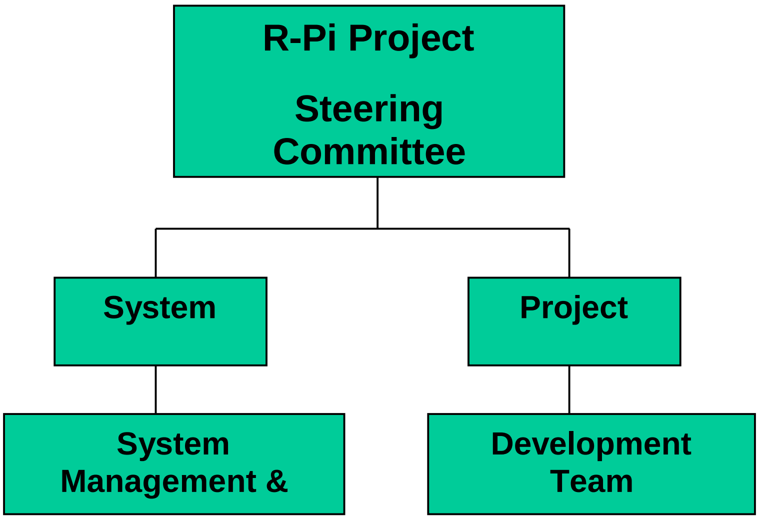
3.2 Evolution of The Plan

In these project, after deciding requirements with all members of group. Everyone take the task which area they can be more beneficial. Every week we had meetings with our project manager and customer. We talk about project and updates that we can or requirements which are done by program. According to these meetings the project completed in progress of time .

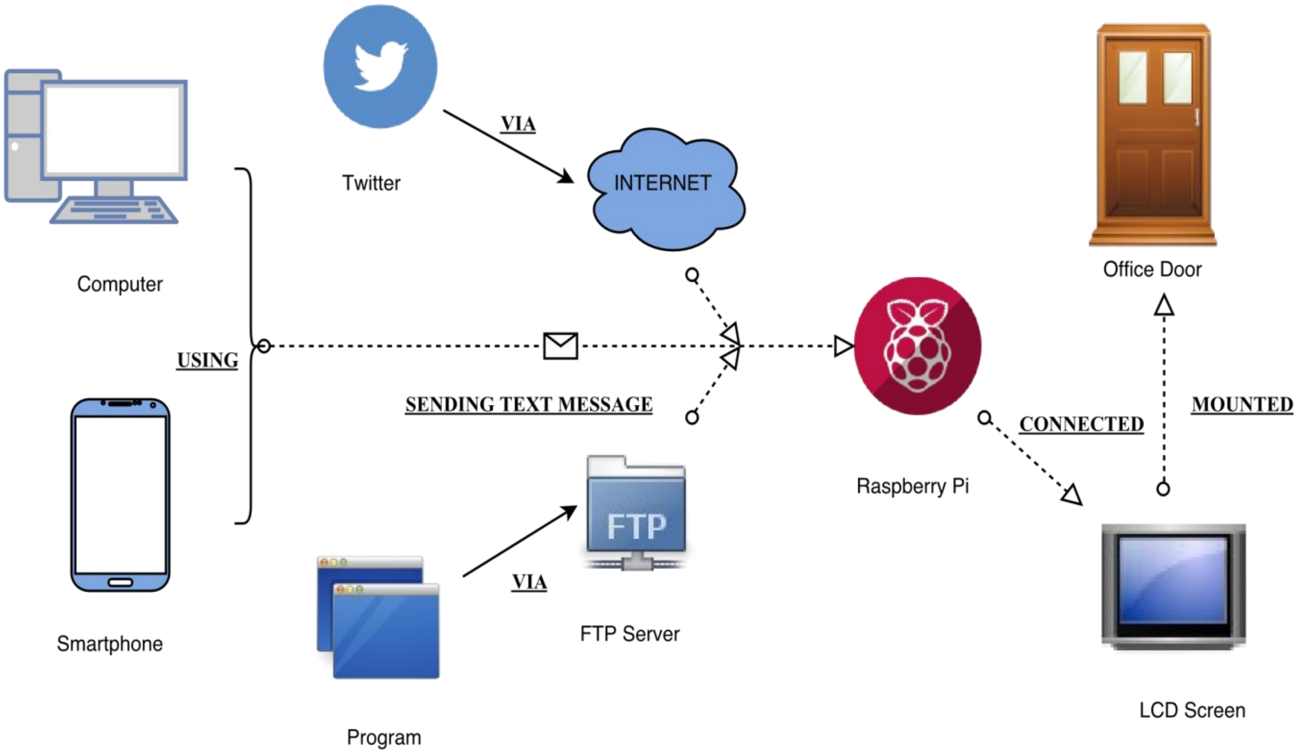
4 PROJECT ORGANISATION

4.1 Overall Structure

Raspberry Pi Project Team: General



4.2 System Architecture



4.3 System Interfaces

External User Interface Requirements

4.3.1.1 User Interfaces

- The user interface for the system will allow the user to easily sending messages to LCD screen. The user should be presented with all main functions on the first user interface page to allow for the user to select the function to use without the need to navigate inward to find it. The interface will need to use It will be accessible through a web interface to allow for data flow and use by various operating systems.

4.3.1.2 Software Interfaces

- The software will need to interface with a case management system to pull data from it and push data updates to it. The connection will be using FTP server and Twitter program which written in JAVA language.

4.3.1.3 Communication Interfaces

- The software will need to interface with a case management system to pull data from it and push data updates to it. The connection will be connection using Internet, FTP server.

5 TECHNICAL PROCESS

5.1 Object Descriptions

5.1.1 Mobile System

5.1.1.1 FTPService = extends from Service
Run()
onStartCommand()
Notification()
readSettings()
onCreate()

5.1.1.2 ListChangePopUp = extends from Activity
onCreate()
Listeners()
writeForSettings()

5.1.1.3 LocationClass = extends from Service
InitialLocation()
Criteria()
setGpsTimer()
writeforSettings()
writeFtpMessage()
readSettings()
onStartCommand()
onCreate()

5.1.1.4 LoginActivity = extends from AppCompatActivity

onCreate()

5.1.1.5 MainActivity = extends from AppCompatActivity

onCreate()

Initialization()

setUpList()

writeForFavor()

writeSettings()

readSettings()

adItemText()

onResume()

readFtpMessage()

openServer()

closeServer()

writeToFtpServer

readFtpFromServer()

writeFtpMessage()

refreshList()

write()

removePast()

Read()

onOptionItemSelected()

onServiceConnected()

5.1.1.6 NetWorkListener = BroadcastReceiver

onReceive()

5.1.1.7 NewsEntry =

Get & Set Methods

5.1.1.8 NewsEntryAdapter = extends from ArrayAdapter<NewsEntry>

GetView

getWorkingView

ViewHolder

5.1.1.9 NewsEntryAdapterForProgram = extends from ArrayAdapter<NewsEntry>

5.1.1.10 NotificationClass = extends from service

readString()

readSettings()

writeForSettings()

readMessage()

writeFtpMessage

writeMessage()

checkEmptyness()

alarmSetter()

calculateFinish()

calculateTime()

5.1.1.11 NotificationEmptyClass = extends from service

onStartCommand

readSettings()

writeFtpMessage

writeForSettings()

5.1.1.12 Program = extends from appCompatActivity

onCreate()
setupTextViews()
setupListViews()
buttonListeners()
setupTabHost()
onOptionItemSelected()
writeForSettings()
startService()
readString()
writeString()
getNewsEntries
prepareListView()

5.1.1.13 Service = extends from Service

5.1.1.14 SettingsClass = appCompatActivity

Preperation()
onCreate()
InitializeLocation()
InitializeWidgets()
checkBoxPrep()
DomainChecking()
onClickListeners()
Permission()
onRequestPermission
writeForSettings()
readSettings()
onResume()
onResumeChanges()

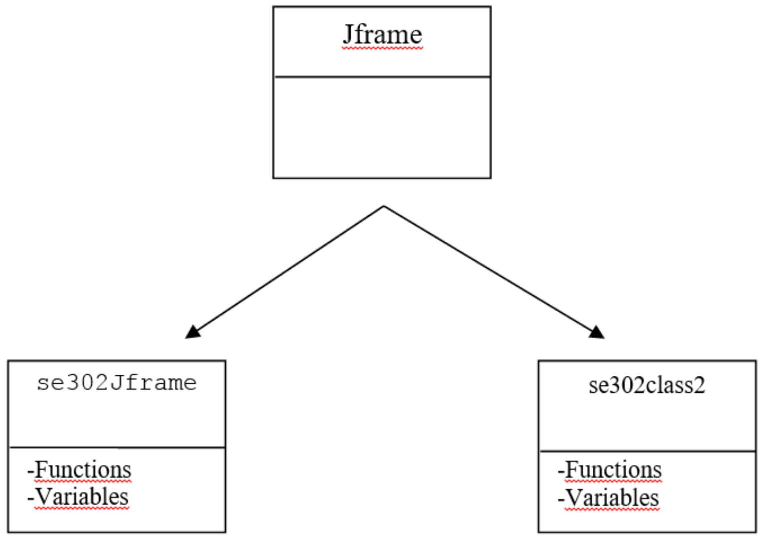
5.1.1.15 VariableClass = extends from AppCompatActivity

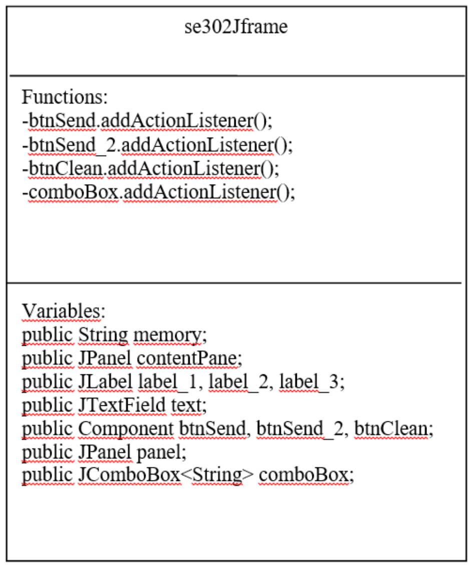
ArrayAdapterResset()
tagsReset()
clearTabtexts()
findTags()
resetAdapterArray()
resetFavorList()

5.1.1.16 FTPPathFinder = extends from AppCompatActivity

onCreate()
Initialization()
getNewsEntries
openFolder()
writeSettings()
closeFolder()
openFTP()
closeFTP()

5.1.2 PC System

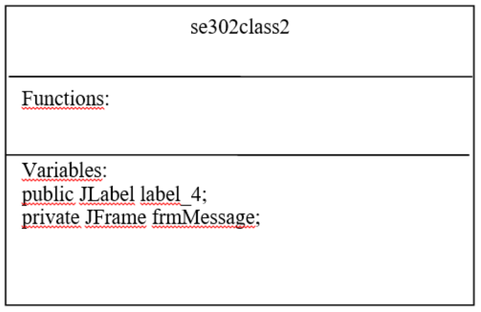




se302Jframe

Functions:
-btnSend.addActionListener();
-btnSend_2.addActionListener();
-btnClean.addActionListener();
-comboBox.addActionListener();

Variables:
public String memory;
public JPanel contentPane;
public JLabel label_1, label_2, label_3;
public JTextField text;
public Component btnSend, btnSend_2, btnClean;
public JPanel panel;
public JComboBox<String> comboBox;



se302class2

Functions:

Variables:
public JLabel label_4;
private JFrame frmMessage;

5.2 Tools

PyQtPython

Java JTK

Android Studio

Raspbian

Crontab

Android SDK

GitHub

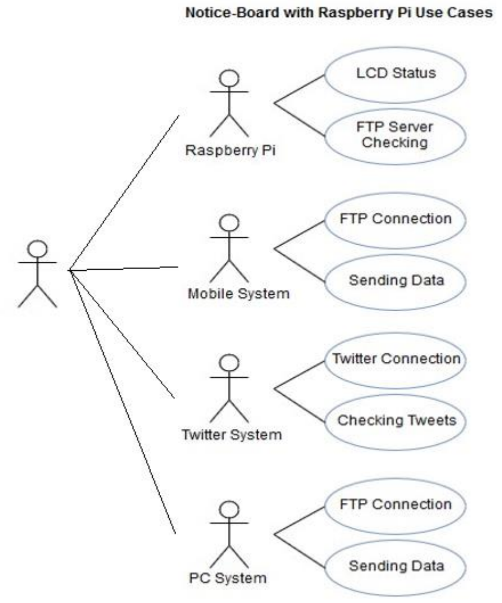
Asana

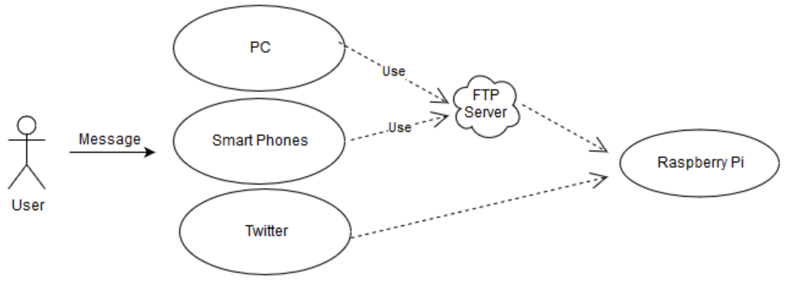
Twitter Account

Eclipse

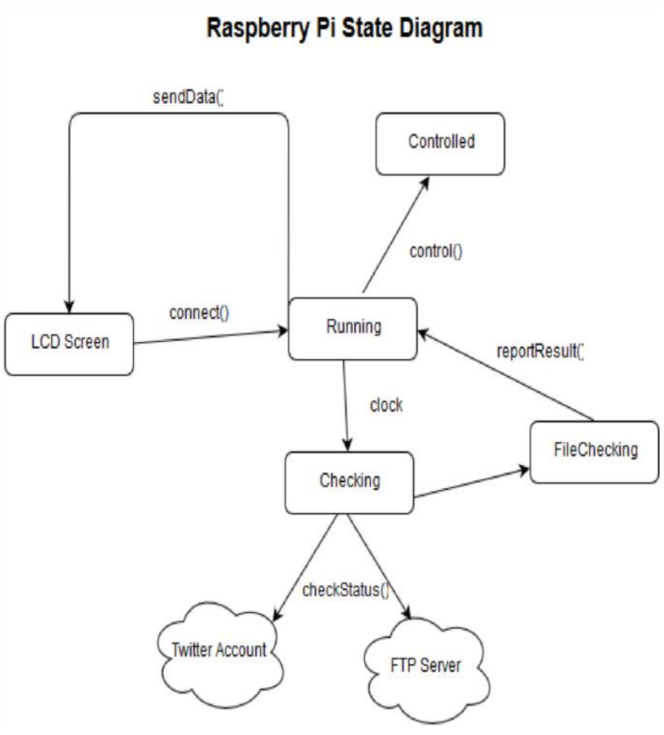
6 PROJECT DIAGRAMS

6.1 Use Cases

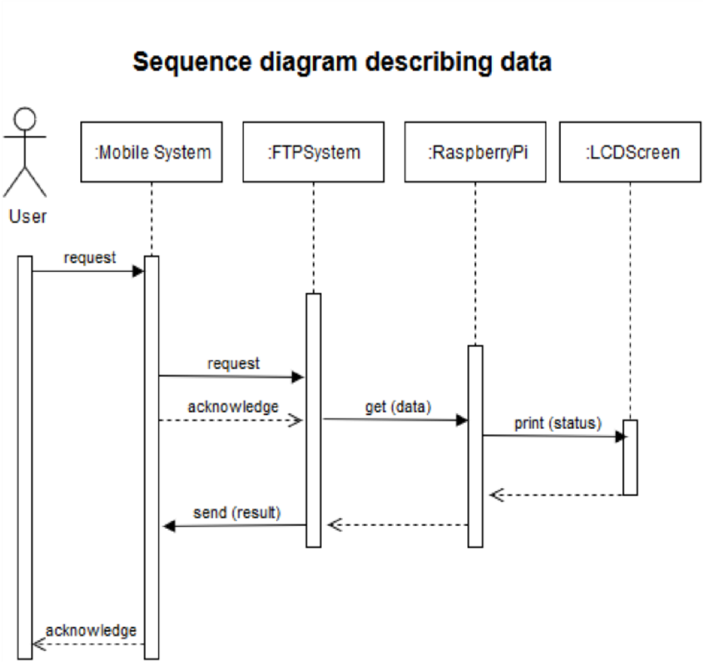




6.2 State Diagram



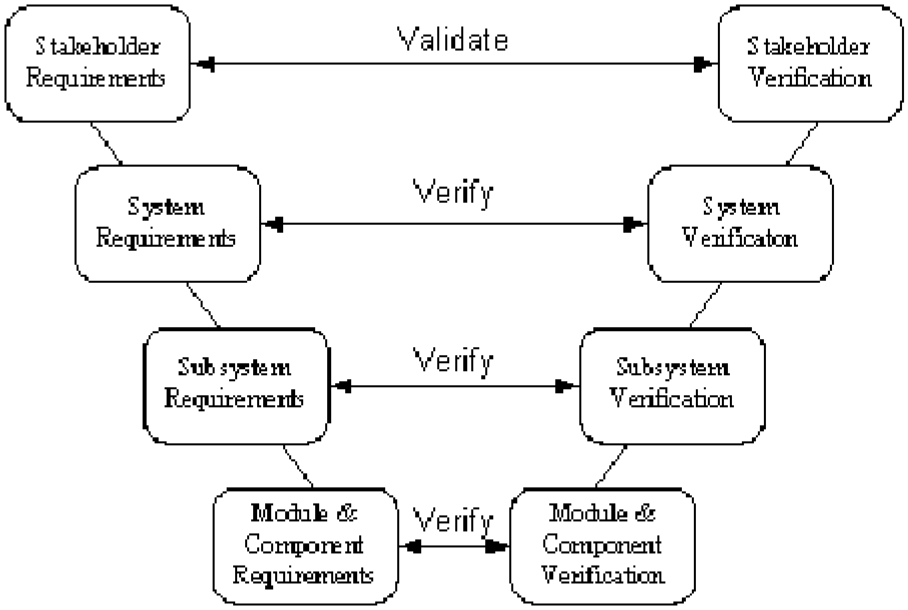
6.3 Sequence Diagram

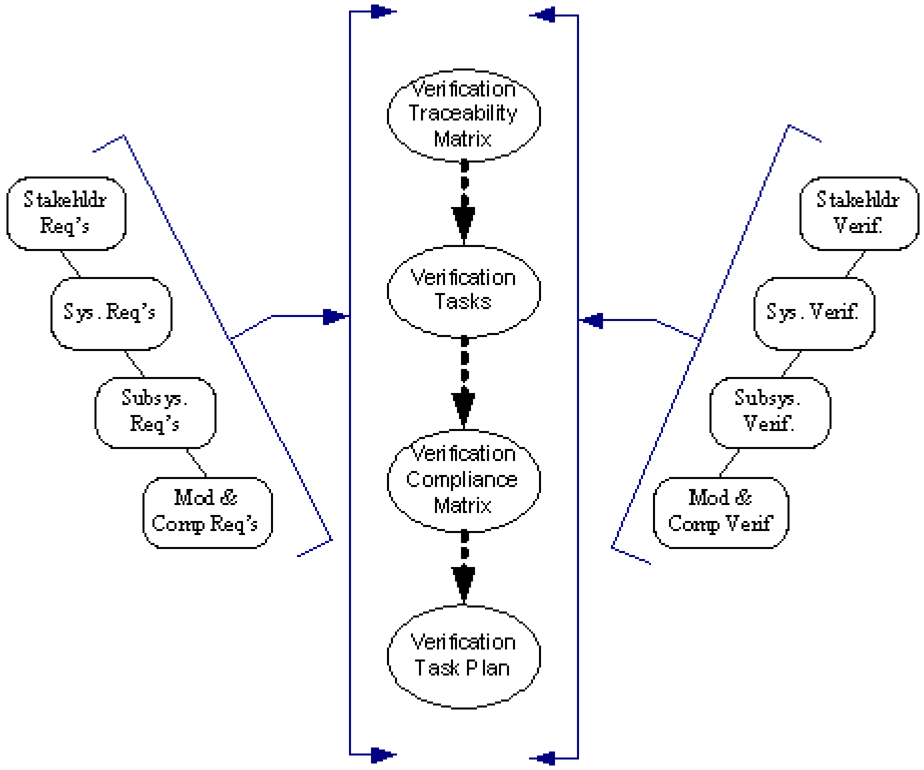


7 PROJECT SCHEDULE

Week	1	2	3	4	5	6	7	8	9	10	11	12
Starting	Oct 14	Oct 21	Oct 28	Nov 4	Nov 11	Nov 18	Nov 25	Dec 2	Dec 9	Dec 16	Dec 23	Dec 30
Requirements Analysis	■	■	■	■								
System & Software Design	■	■	■	■	■	■						
Implementation & Unit Testing			■	■	■	■	■	■	■			
Integration & System Testing					■	■	■	■	■	■	■	
Operation & Maintenance								■	■	■	■	■

8 VALIDATION PLAN





9 PROJECT EVOLUTION

Instead of doing all of these, we can situate raspberry pi in the door and design one page which contains name,date,note etc. students come the office and see the screen if teacher is not there, they write a not. In my opinion, this is so basic and efficient solution.

We think that we could add voice message option to the system or camera. And we develop security of the system to prevent the access of unwanted persons. The system should be personal and confidental. On the other hand rather then using Raspberry Pi with normal pill we can found another option to give power to Raspberry Pi. Because we want more long standing system, changing batteries or problems that user could be faced when the battery is low makes the system unuseful .