Software Engineering Group Project

Interaction and high level design for the system

|  |  |
| --- | --- |
| Author: | Group 03 |
| Config Ref: | SE\_03\_DS |
| Date: | 2015-10-17 |
| Version: | 1.0 |
| Status: | Release |

Department of Computer Science

Aberystwyth University

Aberystwyth

Ceredigion

SY23 3DB

Copyright © Aberystwyth University 2015

CONTENTS

CONTENTS 2

1. Introduction 2

2. Purpose of this Document 2

2.2 Objectives 2

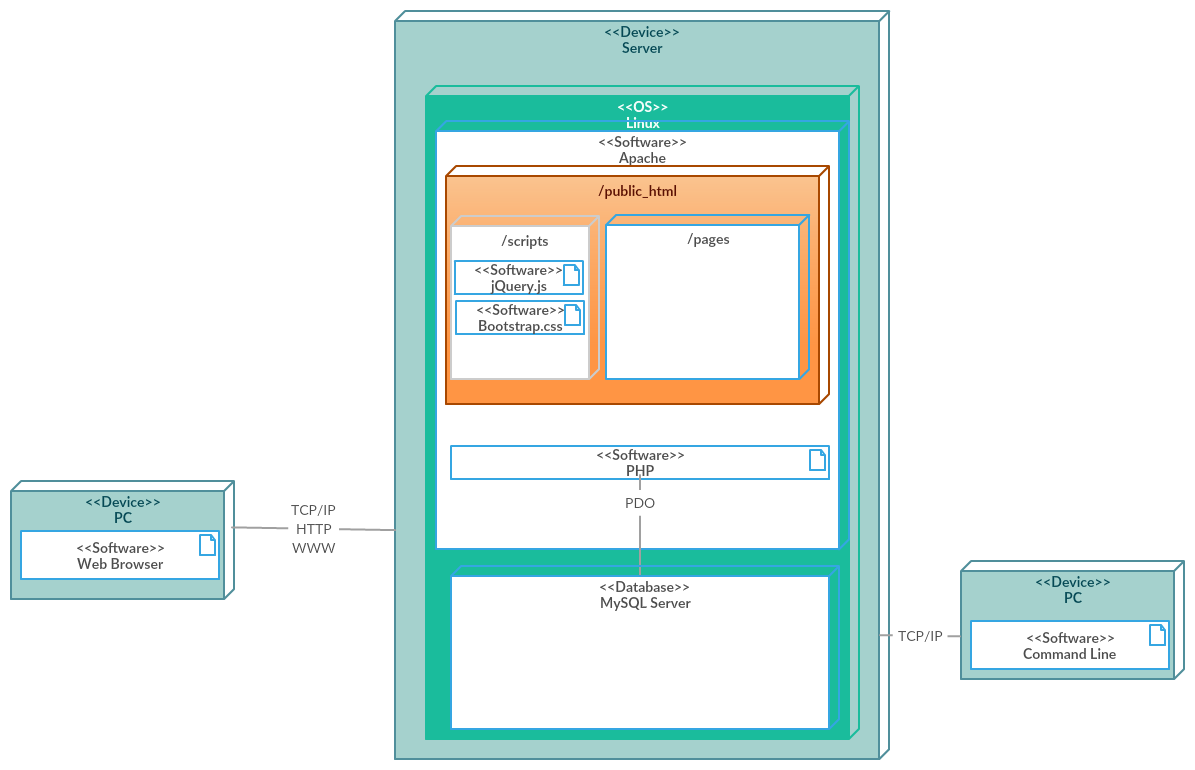
3. BODY OF DOCUMENT 2

REFERENCES 2

DOCUMENT HISTORY 2

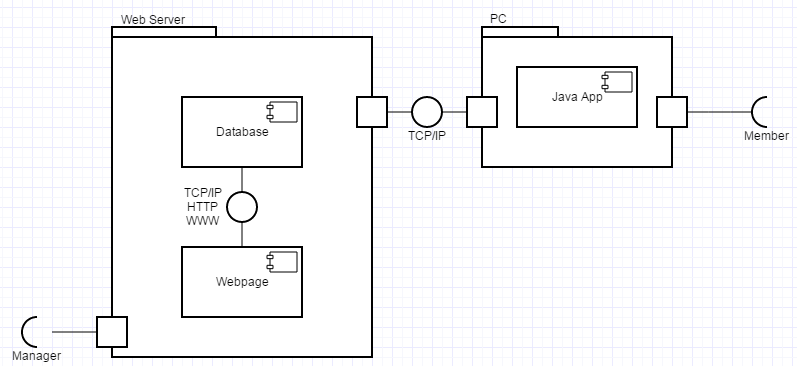
# Deployment description

## Applications in the system



* Web Browser - user’s computer that needs a browser installed and a connection to the internet.
* TCP/IP, HTTP, WWW - data communication protocols that are going to be used to connect to the web server for access to the website (TaskerMAN) and to the database (TaskerSRV).
* Server - details of the server which runs on Linux (aber.ac.uk) that has Apache installed (Web Server) with the PHP extension installed to process PHP files. PHP extension makes calls to the MySQL DataBase(TaskerSRV) via PHP PDO Prepared statements.
* Public\_html - place where we put all the files for the website, including scripts and web pages, which can be viewed via browsers. It can edit the records of TaskerSRV through a web interface using PHP, jQuery and Bootstrap.
* PC - has an Internet connection and TaskerCLI installed on to access TaskerSRV. Java program will use an email for an authorisation to communicate with TaskerSRV using JDBC and MySQL driver.

## Application Interactions



**TaskerMan**

We are using http (Hyper Text Transfer Protocol) for the website. User sends a request to a server asking for a resource (URL (Uniform resource locator)). Resource is a web page (.html or .php or some other file mp3 or image).

User sends a request by typing the URL into the address bar, e.g. www.bbc.co.uk or www.google.com. Once they press enter, browser generates a HTTP request and sends it to the URL.

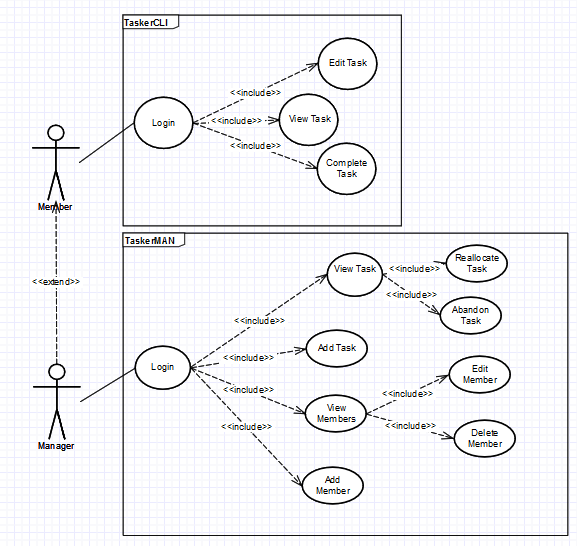
Once the server receives that request, it then responds with the (html/php/mp3) file you asked for, by delivering it to your browser. Chrome/Firefox rendering engine looks at the codes and displays all the links/pictures/text in the way it was intended.

**TaskerCLI**

Java program works by transmitting data packets (TCP Packets) from the user using the program to the MySQL database (TaskerSrv).

# Interaction design

## Use-Case



**Member**

* Member only allow to login TaskerCLI, the desktop application.
* Member allow to view task, a list of assigned works with description.
* Member able to edit task to change the current task details.
* Member able to mark current task as a complete, complete task.

**Manager**

* Manager extends the member and allow to login TaskerMAN, website.
* Manager can view the current task.
* Manager has to view task to reallocate task or abandon task.
* Manager allow to add new task.
* Manager can view group members.
* Manager has to view members to edit members or delete members.
* Manager allow to add new member.

**Example Usage Scenarios**

Complete task

* Member login into TaskerCLI and mark the completed task as complete.

View task

* Member login into TaskerCLI and click on "View Task" button, a list of task will be display.

Abandon task

* Manager has to login into TaskerMAN, then click on the "View Task" button to select which task to abandon.

Add member

* In order to add new member, Manager login into TaskerMAN follow by clicking the "Add New Member" button.

## User interface design

# component description

# Significant classes

## Interface description

# detailed design

## Sequence diagrams

## State diagrams

## Activity diagrams

## Significant data structures

REFERENCES

DOCUMENT HISTORY

| *Version* | *CCF No.* | *Date* | *Changes made to document* | *Changed by* |
| --- | --- | --- | --- | --- |
| 1.0 | N/A | 27/10/2015 | N/A - original version | Group03 |