CA326

Functional Specification

ThinkShift

2020/2021

0. Table of contents

1. Introduction

1.1 Overview

The system we are hoping to implement is a web based scheduling app for a business environment, more catered towards shift based companies such as supermarkets and fast food restaurants. We as a team have had experience with poor scheduling software or lack there of in our various employments so we felt we could drastically improve quality of life for the many part time and shift based employees everywhere. The web based application will allow managers to create employees and issue them a unique ID and pin, they will then be able to check employee availability and create shift “events” for each employee and add them to an interactive day by day breakdown of each week. This will be complemented by a system to allow employees to specify holidays, training days etc

This web based app will strive to provide a clean, colorful interface that is easy to use and welcoming to all, especially for those that are not particularly computer-literate. The app will host a calendar based rota approach, where managers can add times and details, and even use a drag and drop approach to shifts. This web app will be easily accessible, and based on a colour coded system, breaking down shifts into individual departments. Each employee, once the managers have confirmed the rota on a week by week, will receive an email to allow them to view their shifts.

1.2 Business Context

Although we do not have official business backing, this is primarily a business oriented project that could benefit a business tremendously. This can apply to most business contexts in which a timetable for staff is necessary for the business to continue and flourish. We as a group have had experience with poor scheduling systems so we want to create better quality of life for all employees using web technologies.

1.3 Glossary

HTML - HyperText Markup Language

MongoDB - a cross-platform document-oriented database program.

Flask - a micro web framework written in Python

React - A javascript framework for creating web based applications that can dynamically change content and data

2. General Description

2.1 Product / System Functions

Manager Functionality

The manager will be the starting point for this product, they will have to create their own account when starting a new scheduling system for their workplace.

The manager will enter their name and email into the system and they will be given a PIN back by the system. They will then use their email and this PIN to log into the system.

When the manager logs in, they will be able to create employee accounts. The manager will need to know the employee’s name and email address in order to create an employee account. The employee will then be emailed with their PIN and the employee can then log in using their email and PIN.

The manager will then be able to access an interactive schedule where they can set the timetable of every employee. They will also be able to create and assign employees to certain departments. The manager will then be able to *post* this timetable which will make it viewable to the employees and will also notify the employees by email that their timetable has been posted.

The manager will also be able to create events for their employees, such as “training day” or “holidays” to help keep track of how and when to schedule their employees

Employee Functionality

The employees will have a much more limited functionality to the manager. The employee will be able to log in with their email and PIN. Once an employee logs in they will be able to see their calendar which shows their shifts for the current week, and the employee should be able to view shifts for the week(s) ahead assuming the manager has posted the shifts for the week(s) ahead already. Every employee will be notified by email when new shifts have been posted onto the system.

The employee will be able to request time off through the system as well.

Database Related Functions

Only managers will be able to set or change any information on the database, such as employee details or their own details. All timetables posted will also be saved to a database

Web-interface Functionality

The main appearance of our website will be created by integrating HTML with React / Angular js to dynamically change and update our web app. We will use Angular / React to display a calendar of the shifts, both for the employees view and for the managers view (which will be colour coded). The calendars themselves will take up most of the space of the web-interface for both the managers view and the employees view. The manager should be able to schedule their employees shifts with simple drag and drop functionality and the button for the manager to post the schedules should be in a clear and obvious position.

2.2 User Characteristics and Objectives

The user characteristics can be split between two main groups - Employees and Managers. Both sets of users will be expected to have an average understanding of computers but no technical background, however it would be reasonable to expect the managers to receive a brief amount of training on using this system. We can safely assume that both managers and employees alike will understand how to log in and out of the system.

Managers will want a platform where scheduling employee timetables will be an easy and quick task to complete, they will want any creating, editing or deleting of employee accounts to be made as easily as possible. They will also want a system that will automatically tell their employees when a timetable will be posted.

Employees will want a platform where they can simply log in and check their timetable for the week ahead, and occasionally request time off. They will not want something that is overly-complicated or takes a long time to navigate.

We were fortunate enough to speak to a couple of managers who manage shifts for their volunteers and we were able to ask them for their opinion in our project. Most managers could see the benefits of our project which was great to see, the most interesting comments that we heard regularly were the need for a system like this to be able to be changed on the fly easily, and how these managers will typically give their volunteers the same shifts every week, which will be something for us to consider when designing our scheduler.

2.3 Operational Scenarios

Manager Account Creation:

A manager will be presented with a page where they can enter their name and email address in order to create a managers account. A PIN will be emailed to them which they can then use to log into the system as a manager.

Manager Login:

A manager will be presented with a homepage where they can enter in their email address and PIN (given to them by the system) to log in. If they enter invalid details a message will pop up to notify them and ask them to try again, otherwise they will be logged in to their managers account.

Manager Adding Employee:

A manager will be able to add staff by going to a specific part of the web application and entering the employee’s name and email address. The employee will then get an email with their PIN where they will be able to log in as an employee of the manager.

Manager Deleting Staff:

A manager will be able to delete an employee by entering the employee’s name and email address. If the employee’s name and email address do not match, or do not exist in the database then the deletion will have failed and the manager will get a notification to inform them of this.

Manager Setting Shifts:

Managers will be presented with an editable calendar and a list of all their employees and from there will be able to set shifts for all of their employees.

Employee Checking Rota:

Employees will be faced with a page to log in where they enter their email address and PIN. Assuming they log in correctly, the employees will immediately see a calendar with their shifts on it and they will be able to scroll through the timetables week by week.

2.4 Constraints

Time Constraints

The project of course has a very strict deadline of the 12th of March. This could and probably play a factor in what we are able to product for this third year project.

Meeting constraints

Given the current circumstances at the moment, group projects are slightly trickier to coordinate than usual. So meeting up and discussing / explaining our ideas may be trickier than normal

Managerial and Employee UI Requirements

Managers and employees will want the layout of the design in a way that suits their needs. Our website may functionally work, but it will be useless if our users cannot figure out how to use it properly. It will be important for us to make sure that the design of the website suits our potential users, which may take up much of our time to figure this out.

3. Functional Requirements

Managerial Login

Description

The system must contain a database that contains all staff logins so that the correct log in combination of username and pin can be found. All new staff added by the manager will be added to this database in the correct manner.

Criticality

The managerial login can be considered absolutely essential for the web application to work. This is used for identification purposes and is of utmost importance. Once the manager has logged in they can assign shifts for any employee.

Technical issues

The only thing that could be considered a technical issue is to make sure that all managers and staff have a unique login to successfully identify each member of staff.

Dependencies on other requirements

Without the managerial login, no other functions could be carried out such as adding staff to the system or creating and assigning shifts to available members of staff.

Employee Login

Description

This is an extension of the managerial login, but only concerning employees, who can be considered “read only” so to speak. They will be issued a unique username and pin to log in and our database must be able to verify this login.

Criticality

It is also critical for employees to be issued correct and unique login details so they can view their rotas and also confirm their availability.

Technical issues

In a similar vein to the managerial login, the one issue that may arise is if the employee logins are not unique which would cause security/ privacy issues amongst users.

Dependencies on other requirements

This is somewhat critical as employees can confirm or specify their availability so managers can submit shifts into the interactive rota.

Manager adding staff to the employee database

Description

Here the manager is registering an employees details to the staff database, these details include their names and email addresses which will be used as contact references, the employee will then send the employees their unique login credentials upon registration completion.

Criticality

This is also a critical function as it allows shifts to be created and the interactive calendar based scheduler to be populated and usable beyond its start state.

Technical issues

If the manager assigns the wrong information for employees, so employees may not be informed of any shift or rota changes or publications as the wrong email address will be used.

Dependencies on other requirements

This function requires the correct logins for managers and the correct information to input for each employee entry.

Manager adding and publishing rota/schedule to staff

Description

A manager checks employees availability based on multiple factors such as holidays, training days etc. and then creates a scheduler event by adding the employee name, department and time of their shift. This shift will then be added appropriately to the interactive calendar. Once all shifts have been added the manager can publish the rota.

Criticality

This is a critical function as it is how our interactive scheduler is populated with employees and their corresponding shifts, without this our scheduler would be in a constant start state and not updated.

Technical issues

Issues may include conflicting employee information or shift times that may confuse the scheduler or the employees themselves, or shifts are deemed invalid in legal terms e.g (11 hours between shifts).

Dependencies on other requirements

This depends heavily on a manager being logged in correctly and employees being registered with the correct information so that the scheduler can be deemed accurate and current.

User Interface for both managers and staff

Description

This is the main point of contact for both managers and staff. Both will be at first greeted with a login screen, and once either or both parties have logged in, they will be greeted with our main hub of information. This includes a navigation bar that will be different based on manager or staff log in, and our scheduler that will take the majority of the screen real estate.

Within the managerial navbar, there will be links to employee registration and also a create rota link. For employees it will be more limited but employees will be able to click a link to input their availability based on numerous factors.

Once a rota has been published, the scheduler will be populated with color coded and easy to read information and breakdown of each day with shifts and break times for each unique employee.

Criticality

This is a crucial part of our web app. It must be easy to navigate and use for both parties as some members of staff may not be computer-literate. Our scheduler must also be easy to read and follow with simple breakdowns of dayus and department colour codings.

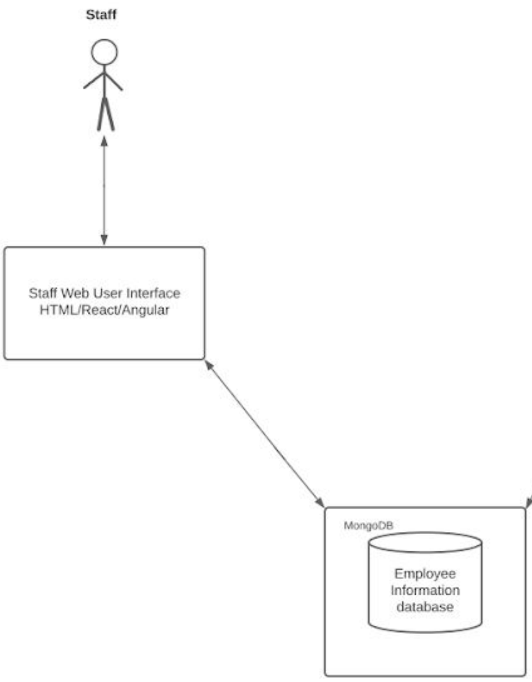
Technical issues

As our user interface relies on react/angular and changes the data based on input in real time, many technical issues could arise within our UI. We must follow our user requirements and make sure our users enjoy using our web app and make their quality of life better within this facet of their employment.

Dependencies on other requirements

This requirement relies heavily on the various requirements thus far, as without managers being logged in and creating shift events for employees, the UI would never be populated with data and would forever be in its start date, complete with an empty scheduler.

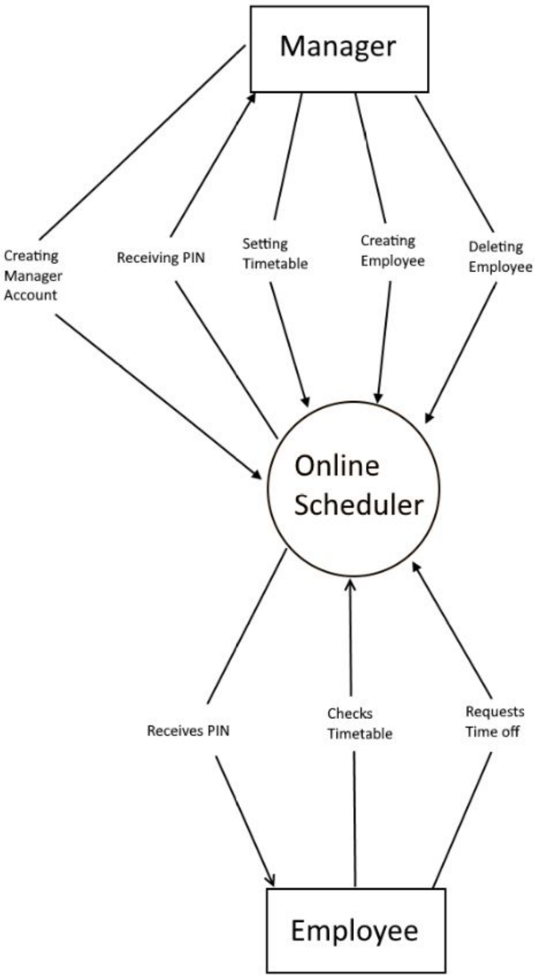
4. System Architecture



5. High-Level Design

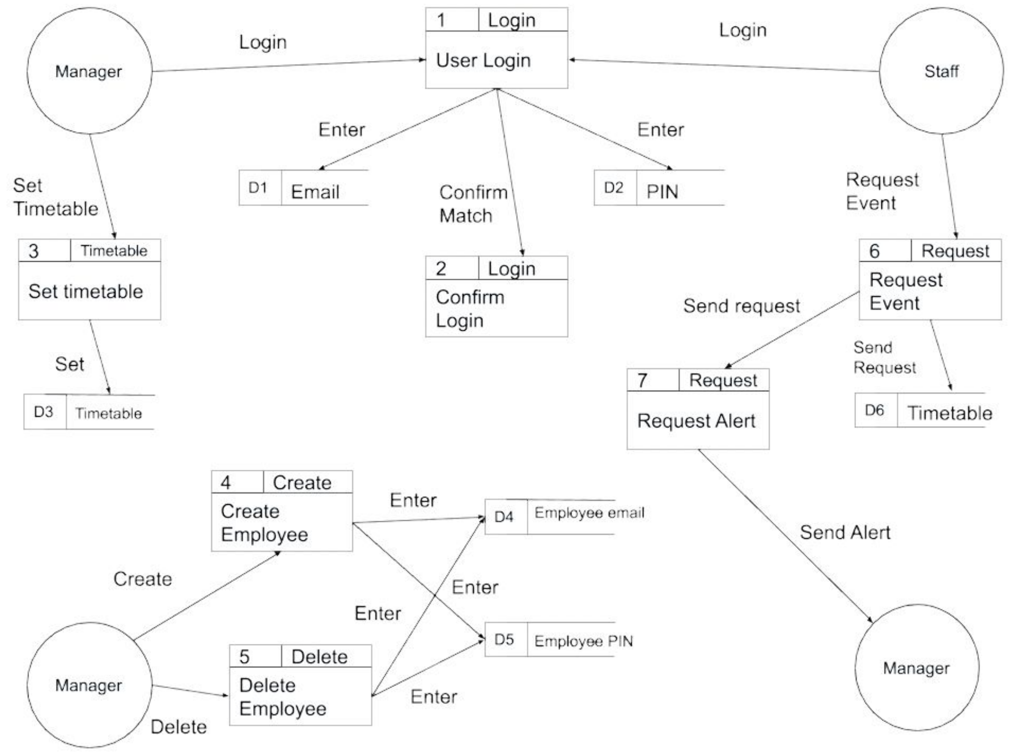
Context Diagram

This context diagram is to illustrate how out web-app will interact with external entities

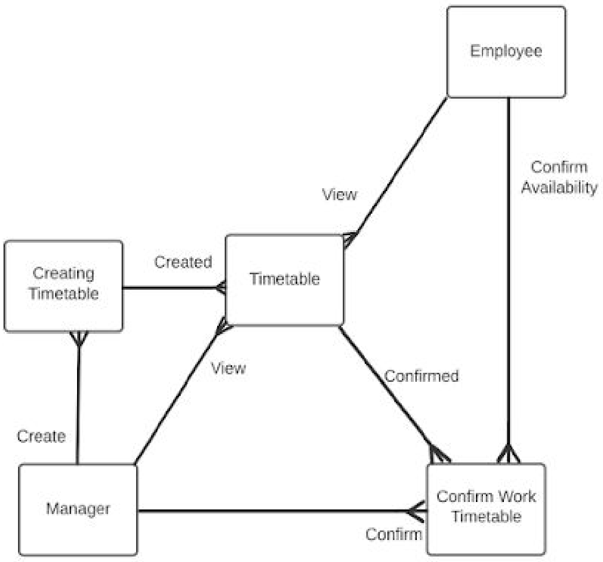


Data-Flow Diagram

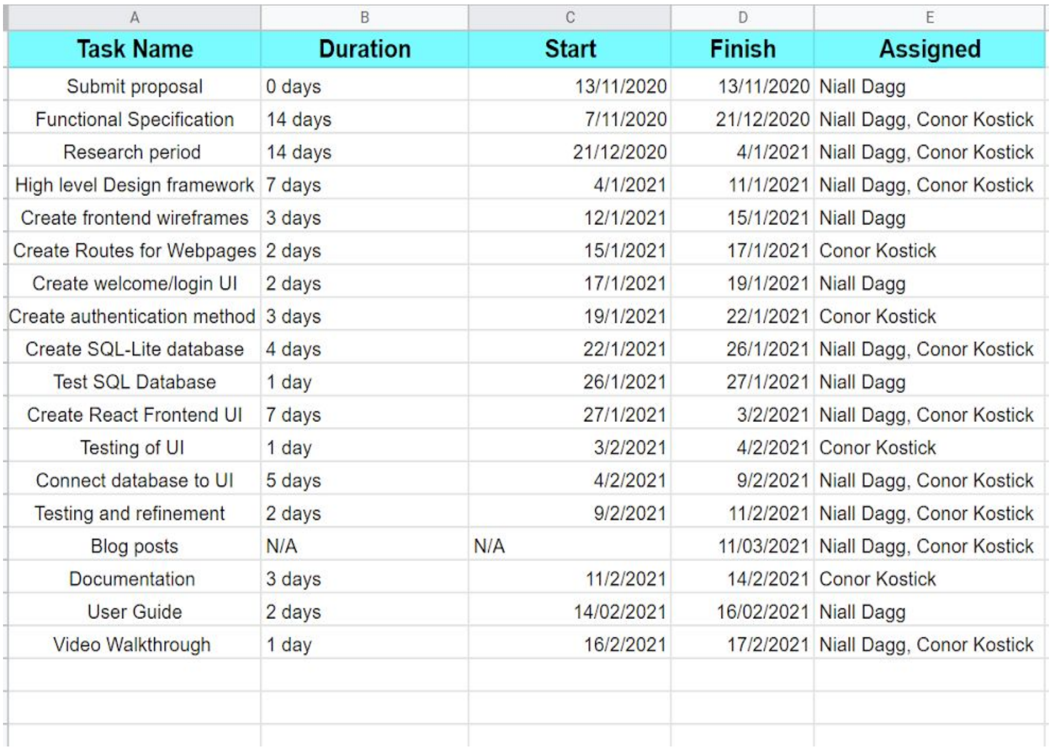
This diagram is to illustrate how data will flow through our web-app and will help give more of an insight as to how the different parts of our web-app will communicate to each other



Logical Data System



6. Preliminary Schedule



A	B	C	D	E
Task Name	**Duration**	**Start**	**Finish**	**Assigned**
Submit proposal	0 days	13/11/2020	13/11/2020	Niall Dagg
Functional Specification	14 days	7/11/2020	21/12/2020	Niall Dagg, Conor Kostick
Research period	14 days	21/12/2020	4/1/2021	Niall Dagg, Conor Kostick
High level Design framework	7 days	4/1/2021	11/1/2021	Niall Dagg, Conor Kostick
Create frontend wireframes	3 days	12/1/2021	15/1/2021	Niall Dagg
Create Routes for Webpages	2 days	15/1/2021	17/1/2021	Conor Kostick
Create welcome/login UI	2 days	17/1/2021	19/1/2021	Niall Dagg
Create authentication method	3 days	19/1/2021	22/1/2021	Conor Kostick
Create SQL-Lite database	4 days	22/1/2021	26/1/2021	Niall Dagg, Conor Kostick
Test SQL Database	1 day	26/1/2021	27/1/2021	Niall Dagg
Create React Frontend UI	7 days	27/1/2021	3/2/2021	Niall Dagg, Conor Kostick
Testing of UI	1 day	3/2/2021	4/2/2021	Conor Kostick
Connect database to UI	5 days	4/2/2021	9/2/2021	Niall Dagg, Conor Kostick
Testing and refinement	2 days	9/2/2021	11/2/2021	Niall Dagg, Conor Kostick
Blog posts	N/A	N/A	11/03/2021	Niall Dagg, Conor Kostick
Documentation	3 days	11/2/2021	14/2/2021	Conor Kostick
User Guide	2 days	14/02/2021	16/02/2021	Niall Dagg
Video Walkthrough	1 day	16/2/2021	17/2/2021	Niall Dagg, Conor Kostick

7. Appendices

Flask - https://flask.palletsprojects.com/en/1.1.x/

MongoDB - https://www.mongodb.com/2

React - https://reactjs.org/