User Document

SmartNotes

User Document

BY

Team Eventually:

Jonathan Fleming

James Gherbaz

Cameron Moon

Calvin Ng

Adam Sanders (Team Leader)

FOR:

College of Science Health and Engineering 2018 CSE3PRA/CSE3PRB

Department of Computer Science at La Trobe University

UNDER THE GUIDANCE OF:

Lecturer: Dr. Torab Torabi

Supervisor: Amin Abkenar

Contents

Copyright

Licensing and Warranty

1.0 - Introduction

1.1 Audience Description

The document aims to inform the two main users of this software system - general staff (employees) and specialised staff (admins) - on how to install and use the software as well as update the system database for admin users.

The regular staff will be the primary user base of this system; they will be viewing and creating meetings, notes etc primarily through the companion app. The employees should only be able to access the meeting information for the meetings that they personally attended unless granted access by an admin. This document highlights how to achieve this.

The admins main role will focus on updating and maintaining the system by adding/removing users from the database, granting access to meetings and approving any system changes. This will enable them to perform extra tasks within the app by granting them access to a separate administration page.

1.2 Applicability Statement

The companion App section on the system is designed to run on any device that has a browser; or in the case of smart phones it can also be launched natively through an app icon.

The raspberry pi interface is designed to run the specific hardware device provided by Arion Technologies it incorporates a Windows OS.

1.3 Purpose Statement

The main purpose of the SmartNotes software system is to provide employees with a simple, quick and intuitive means to create and review various meeting applications, such as notes, tasks, meeting audio etc. It's designed to remove hassle from meetings eg remembering the meeting details, location or deliverables.

The major applications of SmartNotes are:

1. Simplification of an organisations meeting scheduling, creation by localising this information to a single application.
2. Capturing audio of all an organisations meetings to make it easier to refer to or locate specific tasks or deliverables discussed at any point during the meeting.
3. Enabling users to keep track of their attended meetings (past and present) and the specific parts of the meeting agenda that apply to them.
4. Allow users to review and re-listen to meeting audio from any of the meetings they were involved with so they can remember the context in which they made a note or task for themselves.
5. Allow for a transcript of the meeting audio to be generated and reviewed.

1.4 Document Usage Description

The User Document contains the following sections:

1. **Introduction**
   Brief introduction of the system and its purpose and users.
2. **Introductory Kit**
   System features and quick guide.
3. **SmartNote User Manual**
   List of services the app provides and a description of each.
4. **Installation Guide**
   Detailed guide of the steps required to prepare the hardware for this app.
5. **System Administrator Guide**
   Details necessary steps to maintain the system.
6. **Appendices**
   Tables of subsidiary contents.
7. **Bibliography**
   List of source materials.
8. **Glossary**
   Definition of various terms used (focus on technical and domain specific language).
9. **Index**
   Alphabetical list of names, subjects etc with reference to the section of the document in which they occur.

1.5 - Conventions

The conventions used to create the User Document include headings, sizing, spacing, section separation, navigation to help identify the different parts contained within this document. The document was created on confluence using children for each of the categories and subcategories of the document - this allows for simple "one-click" navigation throughout the document and for the creation of content, appendix, table lists.

Every table and figure has a unique number and description so the lists containing these objects is easy to follow and maintain. Confluence does not contain page numbers but instead has section headers; this means that the tables do not list page numbers but instead show which section the figures occur in and can be clicked on to be instantly taken to the relevant section.

2.0 - Introductory Kit

2.1 SmartNotes Overview

SmartNotes is a software solution that hopes to makes it easier for meeting attendees to record notes and tasks. The idea is built around a voice recording device to track meeting conversations. The meetings are recorded through a Raspberry Pi with a 360° microphone and the captured audio will be sent to a central server for processing. A companion app is also planned so that meeting attendees can access this audio and any notes taken during the meeting at their own leisure. The app will be a web app with native smartphone deployment options in order to maximise the user-base and to ensure convenience across all platforms.

The system is broken down into three parts as follows:

1. The front end or interface layer: this is what a user sees and provides the interaction between the user and the system. There are two separate interface layers - one for the companion app and one for the raspberry pi.
2. The Logical layer: this is the series of tasks that can be performed by a user on the system which operates as the connection between the other two layers.
3. The Back end or database layer: this part of the system stores user data, meetings, notes, tasks etc which can be retrieved by the other parts of the system. This layer is deployed on a cloud based service provided by Microsoft Azure.

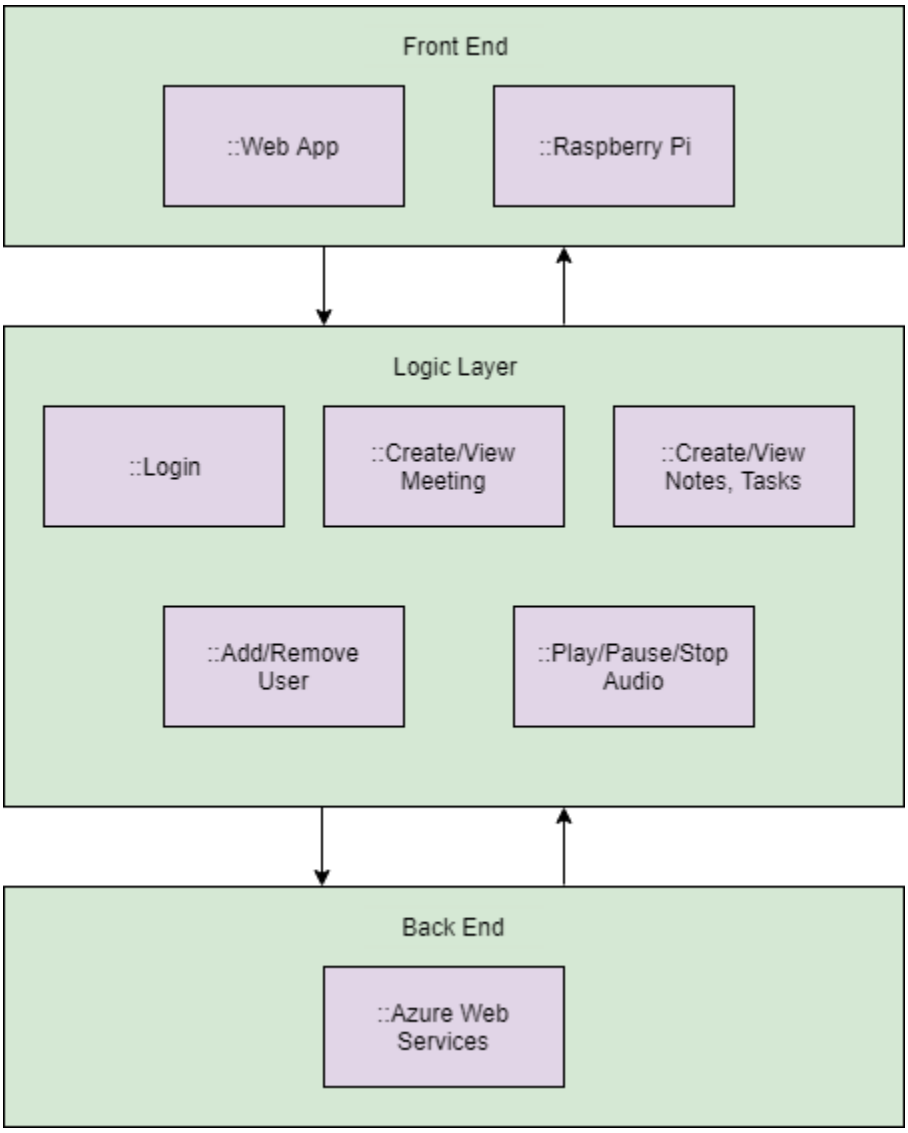

Figure 1. System Overview Design

2.2 Getting Started

This section gives a brief overview of the steps needed for the two different kinds of users to get started. It is a quick reference for the different actions different users can perform. There are two different apps that a user interacts with: a personal companion app and the raspberry pi app.

A general user has many options to get started with this app - they can use any smart phone/tablet, PC, laptop, or device with internet to access the companion app. Smartnotes is not currently deployed to any particular website but once up and running the user simply goes to the site and use or create their login information. After successful validation the main page is displayed, from here all functionality can be performed. A smart device user may also choose to download the app and then an icon will appear on their devices home page and app list - they can choose to launch the software in this manner if they so choose. Both will require valid user credentials.

General users may also be required to use the raspberry pi on occasion; the software is pre-installed and automatically boots up so the user need only navigate the simple menus on the pi app. The pi app is primarily for use during a meeting and users are advised to keep the embedded system in the desired meeting room.

An admin user is responsible for the maintenance of the software system and as such will require competency in database management; this document will instruct the admin how to deploy the app through azure portal and also detail how to maintain the database (primarily add/remove users) from the database.

This is a list of the tasks users are able to perform in the two different SmartNotes systems:

Companion App:

1. User login
2. Create meeting
3. Take notes
4. Join a meeting
5. Add user to completed/past meeting
6. Add user (Admin)
7. Remover user (Admin)
8. View past meeting
9. Play past meeting audio
10. Pause past meeting audio
11. View previous meetings notes

Rasperry Pi:

1. Fast create meeting
2. Start meeting
3. Stop meeting
4. Pause meeting
5. Resume meeting

An in depth look at each of these can be found in the next section of this document.

3.0 - SmartNotes User Manual

This section gives the users a comprehensive view and guide for all of the screens that they may encounter in this system. There are 12 screens in total; the screens allow a user to complete all of the tasks outlined in the introductory kit.

The following table is a list of the different screens within the SmartNotes main app.

#	Service/functionality
1	User login
2	Home screen
3	Create meeting
4	Take note
5	Join a meeting
6	Past meeting list
7	Upcoming meeting list
8	Play past meetings

Table 1. App Functionality

The following table is a list of the different screens within the SmartNotes raspberry pi app.

#	Service/functionality
9	Home screen (Pi)
10	Fast create (Impromptu meeting)
11	Scheduled meeting pin entry
12	Current meeting

Table 2. Pi Functionality

3.1 User login

This is the first screen a user is met with when first loading up the companion app; it is a very simple screen which enables a user to login so that they can access their relevant meeting data. Figure 2 is the wire-frame of the login screen.

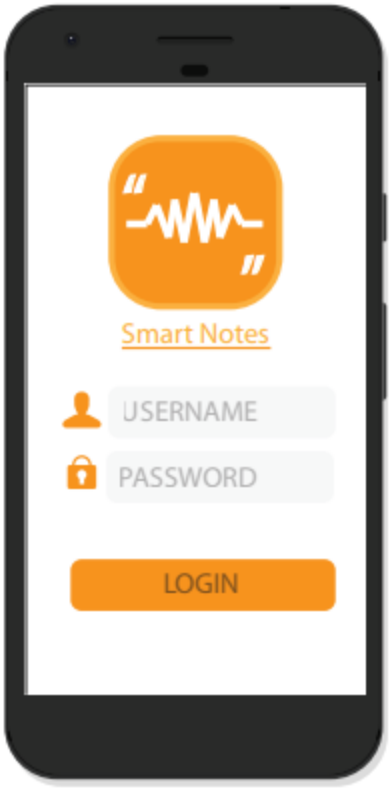

Figure 2. Companion App Login Screen

There are three possible actions here:

1. Enter valid credentials - this will take the user to the apps main page. This page is detailed in section 3.2
2. Enter In-valid credentials - an error message window is pushed to the screen informing the user of an unsuccessful attempt
3. Exit app - the app fully closes down

3.2 Home page

This is the main screen of the companion app; from here navigation to all menus is possible. The persons user name is displayed across the top of the screen and their personalised list of upcoming meetings is shown. Only meetings in which the have entered a valid code will be displayed here.

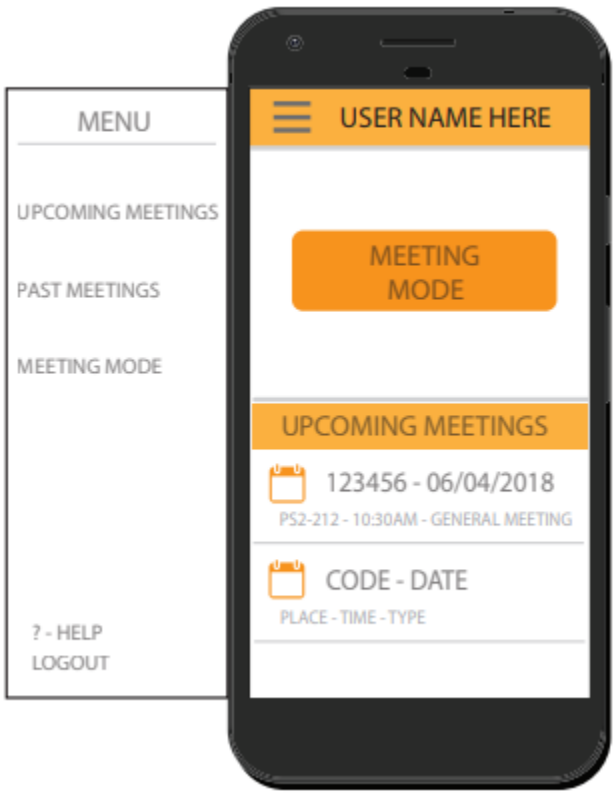

Figure 3. Companion App Home Screen

There are many actions which a user can take at this screen:

1. From the menu screen:
    a. Can select upcoming meetings - will take the user to their upcoming meetings list. This page is detailed in section 3.7
    b. Can select past meetings - will take the user to their previous meetings list. This page is detailed in section 3.6
    c. Can enter meeting mode - will take the user to the join meeting screen. This page is detailed in section 3.5
    d. Can enter help sub-menu - will pop up information text box when any element on the home page is clicked and a brief message will display to inform the user of the buttons function
    e. Can log out - will take user back to the login screen.
2. Can enter meeting mode (bypassing the menu screen) - will take the user to the join meeting screen. This page is detailed in section 3.5
3. Can select upcoming meetings (bypassing the menu screen) - will take the user to their upcoming meetings list. This page is detailed in section 3.7
4. Exit app - the app fully closes down.

3.3 Create meeting

This section enables a user to plan an upcoming meeting. The meeting automatically generates a unique code, which then enables other users to access this meeting. For security reasons there is no way to publish/share this code in the app, users must distribute it externally. The time stamp is also automatically generated (the app uses server time to ensure the database and pi are synced). A date for the upcoming meeting is the only other field with must be entered. A meeting time, location and time of meeting can also be entered but are not needed for this app.

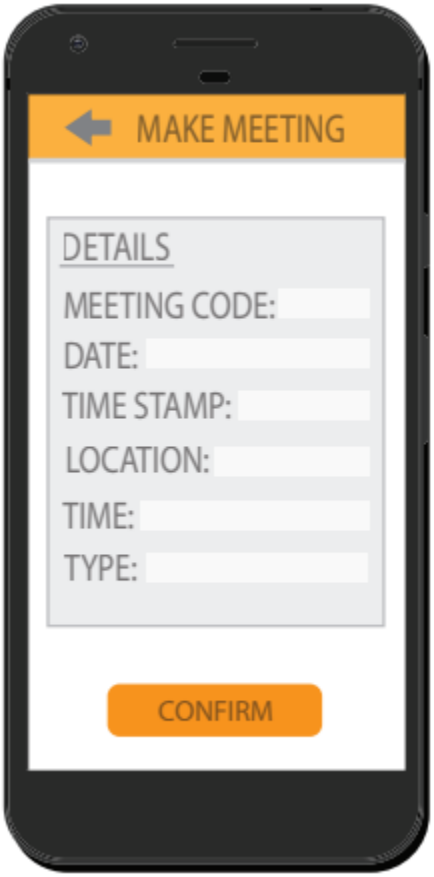

Figure 4. Companion App Create Meeting Screen

There are four possible actions here:

1. Enter valid meeting details - this will create a meeting and return the user to the home page.
2. Enter in-valid meeting details (no date) - an error message window is pushed to the screen informing the user of an unsuccessful attempt.
3. Select 'back' button - will take user to the main menu.
4. Exit app - the app fully closes down.

3.4 Join a meeting

In order to join a meeting two conditions must be met: The meeting code must exist (be valid) and the meeting must currently be in progress (from the pi app). If these conditions are met the meeting mode is enabled (that screen is shown in section 3.5)

Figure 5. Companion App Join Meeting Screen

There are four actions which a user can take here:

1. Enter valid meeting code - this will take the user into 'meeting mode'. This page is detailed in section 3.5
2. Enter in-valid meeting code - an error message window is pushed to the screen informing the user of an unsuccessful attempt.
3. Select 'back' button - will take user to the main menu.
4. Exit app - the app fully closes down

3.5 Meeting mode

This screen is shown when a user successfully joins a meeting (ie enters a the correct meeting code and the meeting is in progress). It enables a user to make notes at any given point in the meeting. The timer displays how long the present meeting has been going (again this is server time to ensure syncopation).

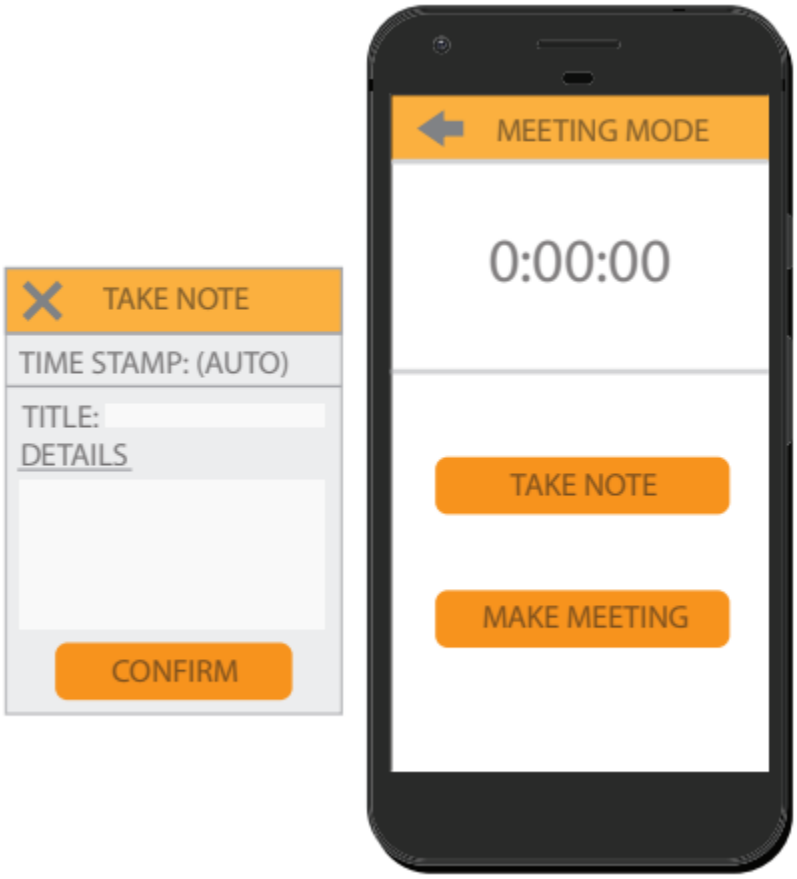

Figure 6. Companion App Meeting Mode Screen

There are four options in the meeting mode screen:

1. Take note - this enables a pop up window in which a user can create a note, the note automatically takes the current meeting time so the user can access the relevant bit of audio quickly. There are two options for the user:
    a. Input the title and details of the note and hit confirm to finalise the note.
    b. Hit cancel - the note is discarded at this point.
2. Make meeting - a make meeting box opens, used to schedule meetings from within a current meeting. The functionality is exactly the same as section 3.3
3. Select 'back' button - will take user to the main menu.
4. Exit app - the app fully closes down

3.6 Past meeting list

This page lists all of the previous meetings a user has attended, from here they can also navigate to that meetings audio and notes they created for that meeting. They are displayed in order of occurrence (with the most recent meetings at the top).

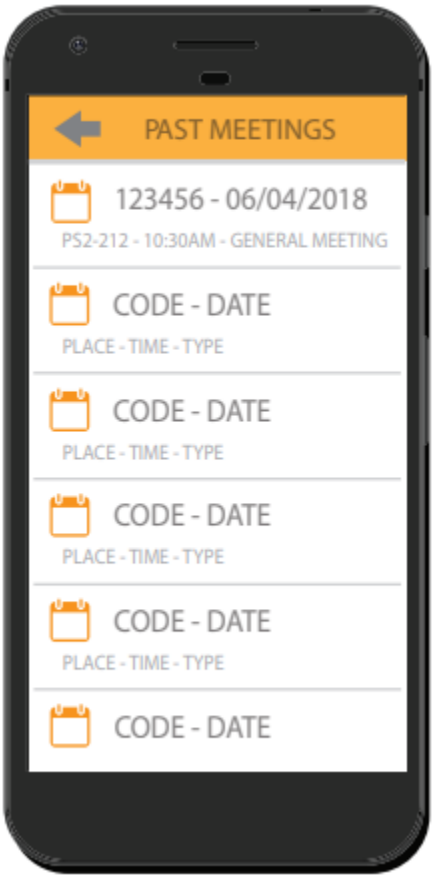

Figure 7. Companion App Past Meeting Screen

There are three possible actions here:

1. Select a past meeting - will take user to the meeting playback screen. This page is detailed in section 3.8
2. Select 'back' button - will take user to the main menu.
3. Exit app - the app fully closes down.

3.7 Upcoming meeting list

This page lists all of the upcoming meetings a user has added, from here they can also navigate to that meetings details. They are displayed in order of occurrence (with the most recent meetings at the top).

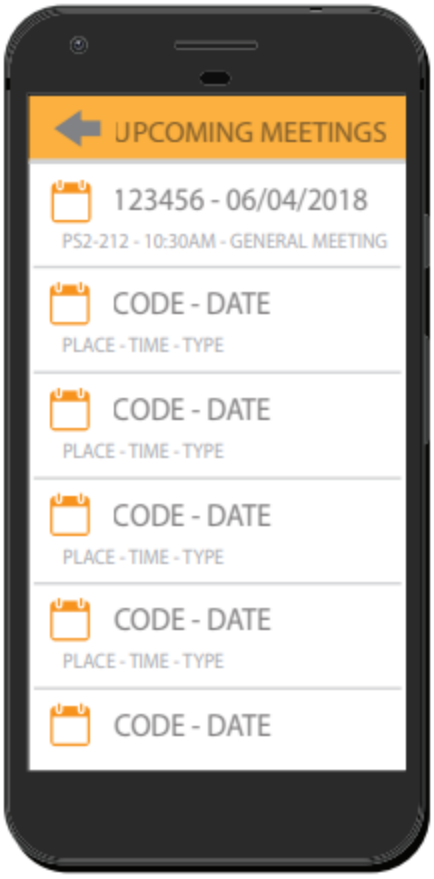

Figure 8. Companion App Upcoming Meeting Screen

There are three possible actions here:

1. Select a meeting - will take user to the meeting information screen. This page is detailed in section 3.4
2. Select 'back' button - will take user to the main menu.
3. Exit app - the app fully closes down.

3.8 Play past meetings

This screen allows a user to playback the audio from a previous meeting they have access to. It also displays the notes that the user has made. It is a typical audio playback page; a scrolling waveform, play/pause, skip and fast fwd/rwd are all implemented to help the user navigate the meetings audio. Meetings can go for a long time and can have many irrelevant topics of discussion for the users so implementing as many time saving features was important.

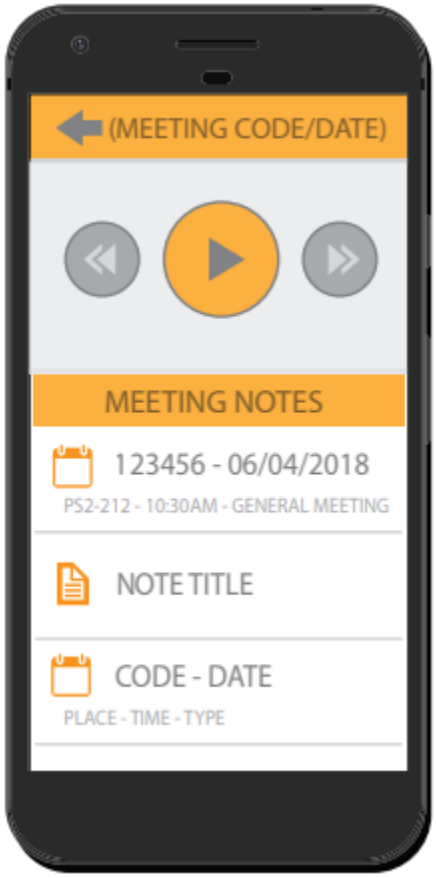

Figure 9. Companion App Play Meeting Screen

There are many possible actions here:

1. Select play - will start to play the meeting audio.
    a. Select pause - will pause the meeting audio - these buttons alternate.
2. Select any point on the waveform - the audio will move to that section.
3. Select back or forward arrows - this skips the audio file 5 seconds for each press.
4. Select a note - audio will move to this point and display the note in a pop up menu.
5. Select 'back' button - will take user to the main menu.
6. Exit app - the app fully closes down.

3.9 Home page (Pi)

This is the main screen of the raspberry pi app; from here navigation to all menus is possible. Two types of meeting can begin here: a scheduled and an impromptu meeting, allowing users to have unplanned meetings is an important feature of the app.

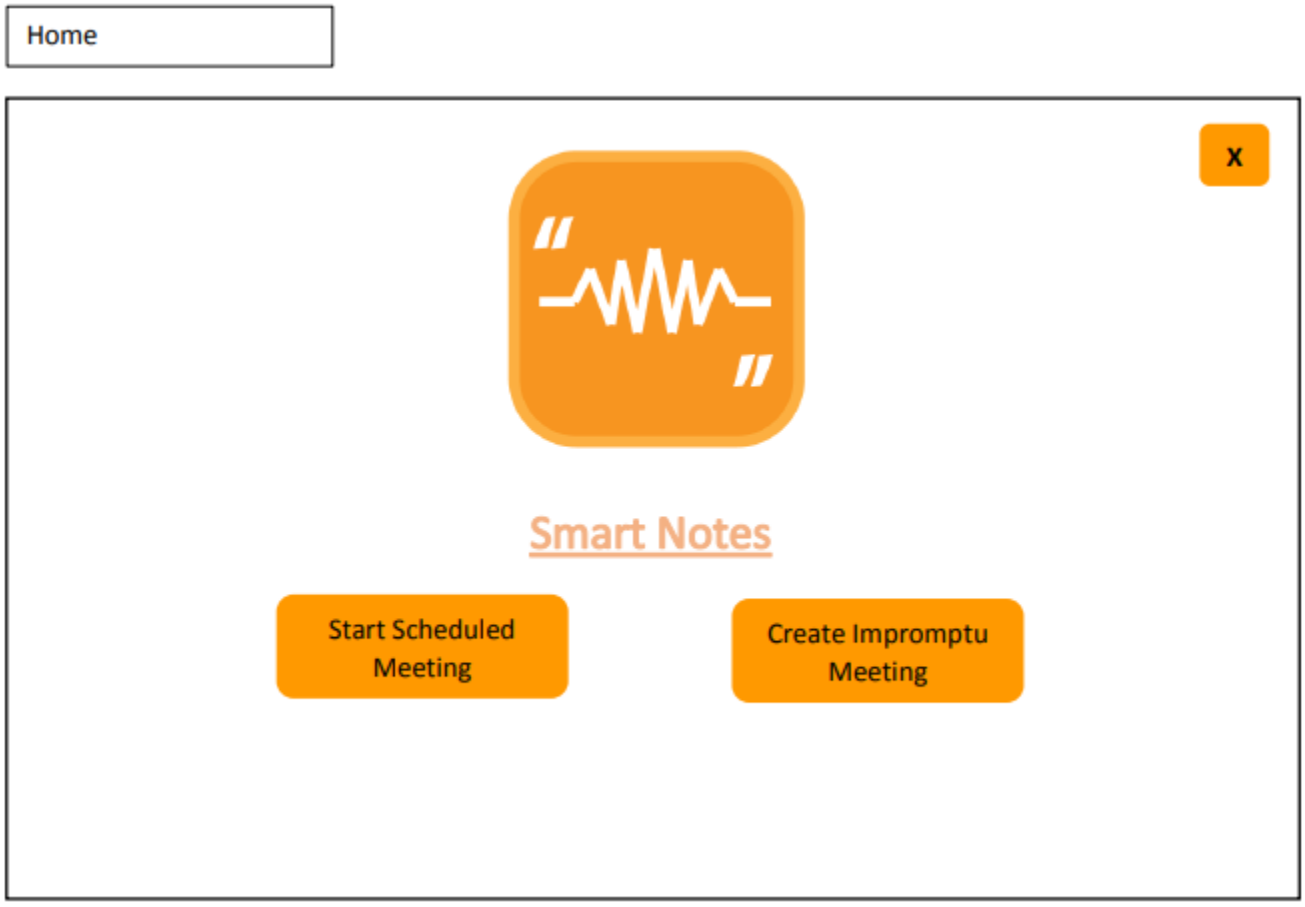

Figure 10. Pi App Home Screen

There are three possible actions here:

1. Start scheduled meeting - will take the user to the code selection screen. This page is detailed in section 3.11
2. Create impromptu meeting - will take the user to the current meeting screen. This page is detailed in section 3.12
3. Exit app - the app fully closes down.

3.10 Fast create (Impromptu meeting)

The impromptu meeting is for when a meeting is needed but the users have not scheduled a meeting previously. The screen waits until at least one person to join before moving onto the current meeting screen, begin recording audio, and allowing users to take notes.

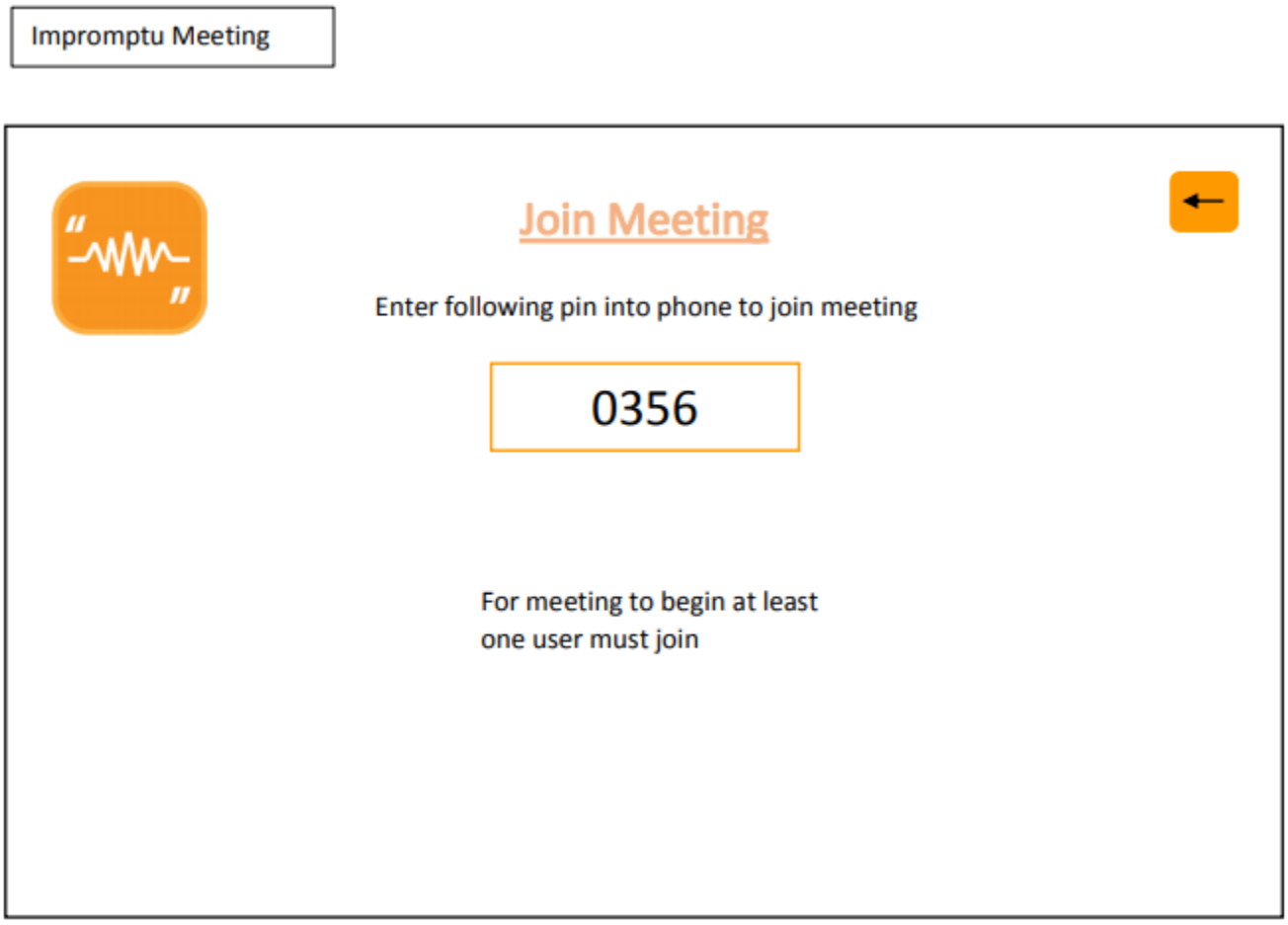

Figure 11. Pi App Fast Create Screen

There are four possible actions here:

1. A user (through the companion app) joins the meeting - this will begin the meeting.
2. No-one joins the meeting - the app will time out after 5mins and return to the main page
3. Select 'back' button - will take user to the main menu.
4. Exit app - the app fully closes down.

3.11 Scheduled meeting

This is the screen in which users enter a planned meeting code. If the code is valid the meeting will begin and users can join the meeting and begin taking notes and

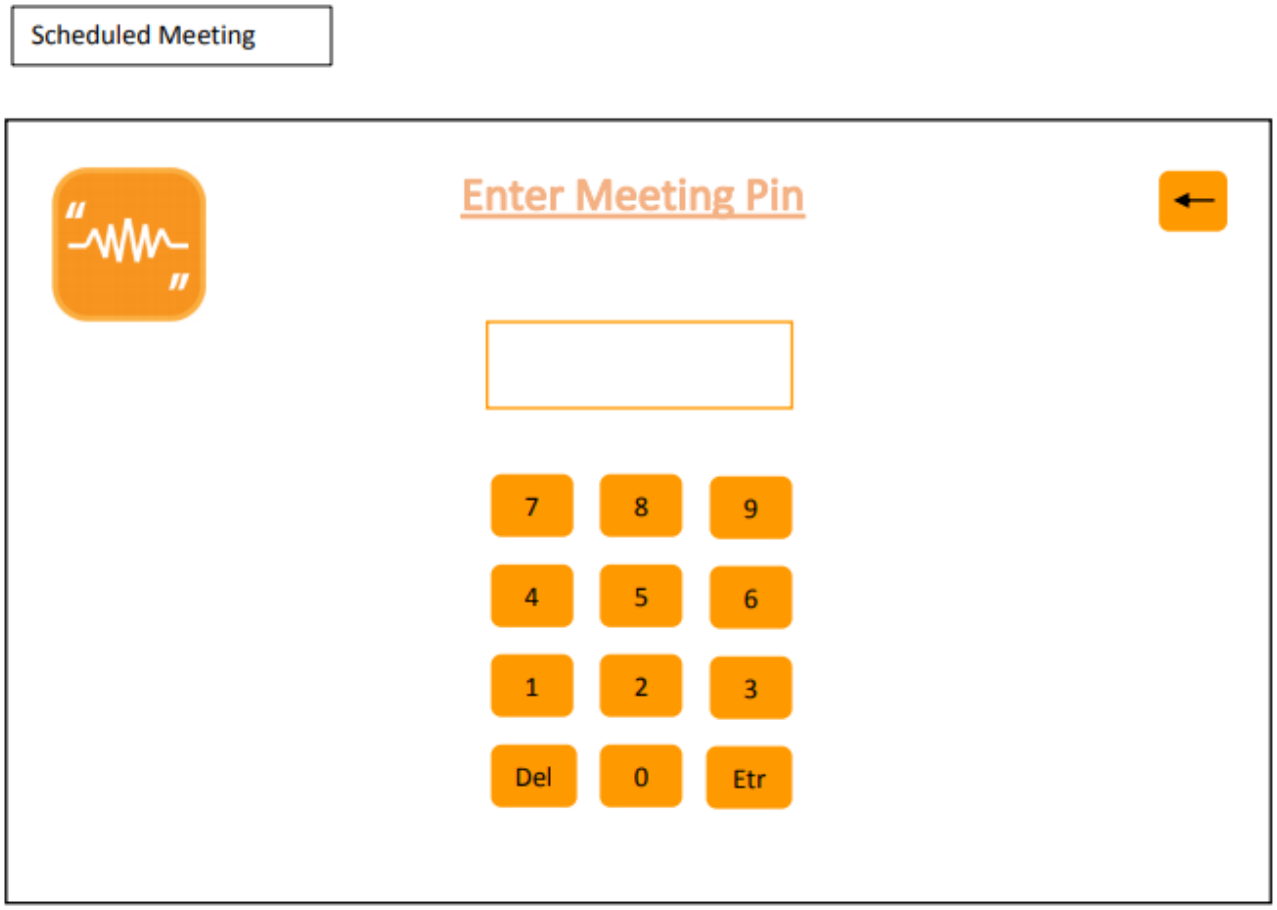

Figure 12. Pi App Scheduled Meeting Screen

There are four possible actions here:

1. Enter valid meeting code - this will start a meeting and display the current meeting screen. This page is detailed in section 3.12
2. Enter in-valid meeting code - an error message window is pushed to the screen informing the user of an unsuccessful attempt.
3. Select 'back' button - will take user to the main menu.
4. Exit app - the app fully closes down.

3.12 Current meeting

The current meeting screen displays all of the current meeting information for the meeting that is in progress. The audio is recording at this point and users are able to join and make notes through their companion app. When the meeting is over as soon as the 'stop' button is pressed, at this point users can no longer create notes. The screen also allows users to pause the recording - for breaks, of the record discussions etc

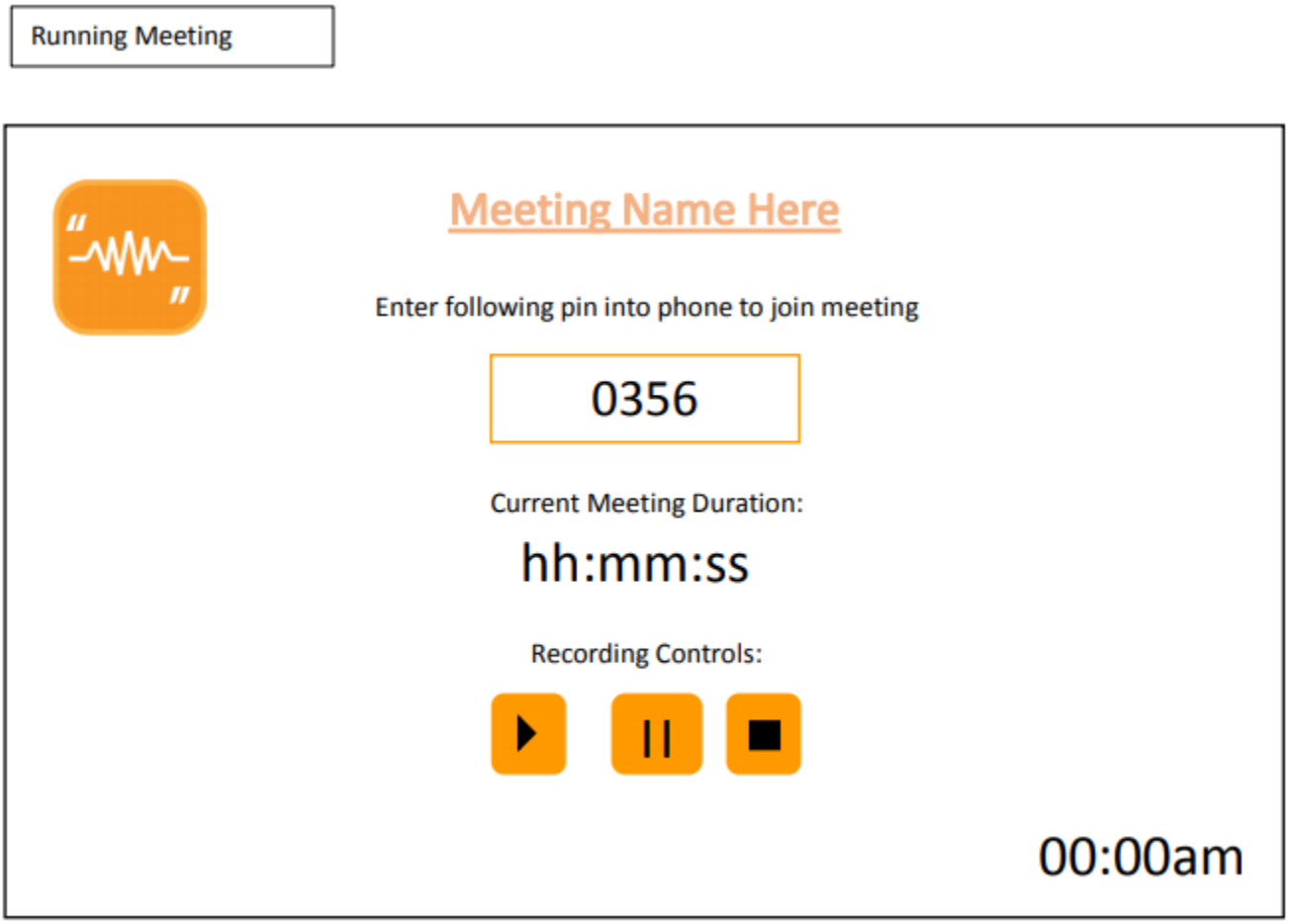

Figure 13. Pi App Current Meeting Screen

There are three possible actions here:

1. Pause current meeting - mutes the microphone so that no audio is recorded until the resume button is pressed.
2. Resume current meeting - will un-mute the microphone given that the meeting is paused; will display an error message if the meeting is not paused.
3. Stop current meeting - the meeting is concluded and all users are returned to their home screens on the companion app and no more notes can be taken. The meeting will now show up in the previous meetings list. A dialog box will pop up asking if the user is sure they want to end the meeting to reduce the chances of accidental meeting ending.

4.0 - Installation Guide

The companion app and database app is fully contained and needs to be deployed to some sort of online service. The System Administration Guide gives a detailed instructions how to deploy the project to Azure web services - but any other (similar) online deployment service would also be permissible.

The raspberry pi does need some technical expertise as it has embedded hardware (microphone array) which is not a standard piece of hardware. Drivers must be installed in order to use the microphone and the software must be pre-loaded onto the device itself. Fortunately this means that no further development is needed on the part of this hardware (except for updates of course - which will require manually updating the device). Once installed the software automatically boots up whenever the device is turned on.

Required libraries and software for the pi app:
for UI:

- tkinter – usually preinstalled with python
- ImageTk.py
- Tmage.py

for web connectivity:

- json.py
- requests.py

for audio playback and conversation:

- pyaudio.py
- wave.py
- pydub - https://github.com/jiaaro/pydub#installation
- ffmeg - apt-get install ffmpeg libavcodec-extra

Users do not need to to install any software to access this system. But they can download the app onto their smart devices if they prefer to deploy from tan icon on their devices homepage rather than through a browser.

5.0 - System Administrator Guide

Here is a six step guide to deploying the app to Azure services. Any cloud based deployment service (such as AWS) would also be sufficient too.

Step 1) Set up an azure account or log into an existing one.

Step 2) Create your project resources you will need the ones shown below.

NAME	TYPE	RESOURCE GROUP	LOCATION
nahfam	SQL server	SmartNote1	Australia Southeast
Nahfam	App Service plan	SmartNote1	Australia Southeast
FamNah (nahfam/FamNah)	SQL database	SmartNote1	Australia Southeast
SmartNote1	App Service	SmartNote1	Australia Southeast
smartnotesstorage	Storage account	SmartNote1	Australia Southeast

Figure 14. Project Resources

Step 3) Navigate to you App Service and click get publisher profile.

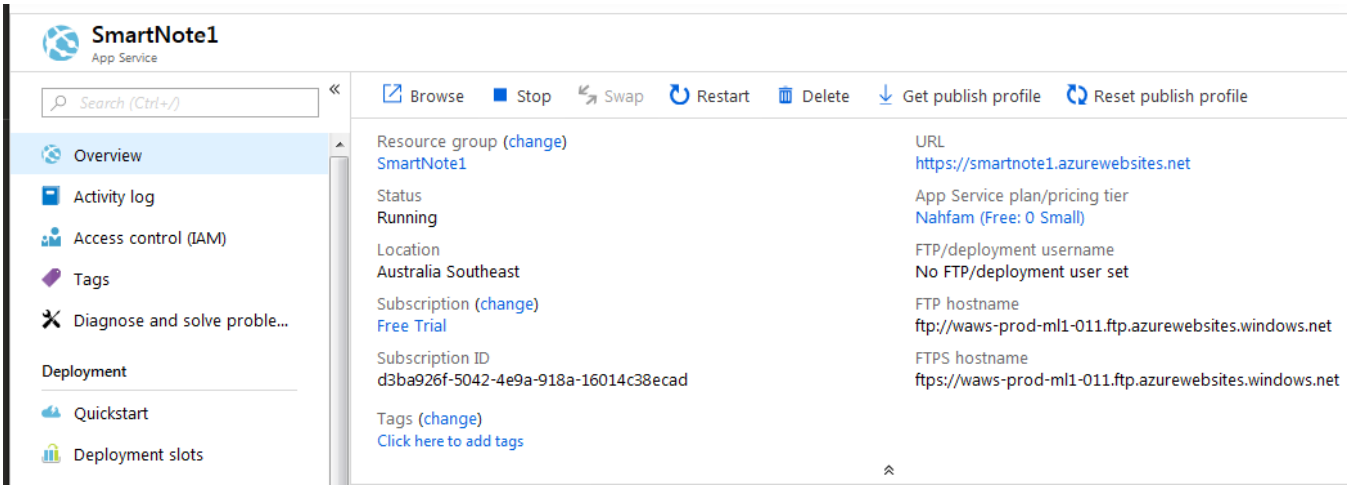

Figure 15. App Service

Step 4) In your project select Build > Publish then under publish select New Profile > import profile

Navigate to your downloaded profile and select it.

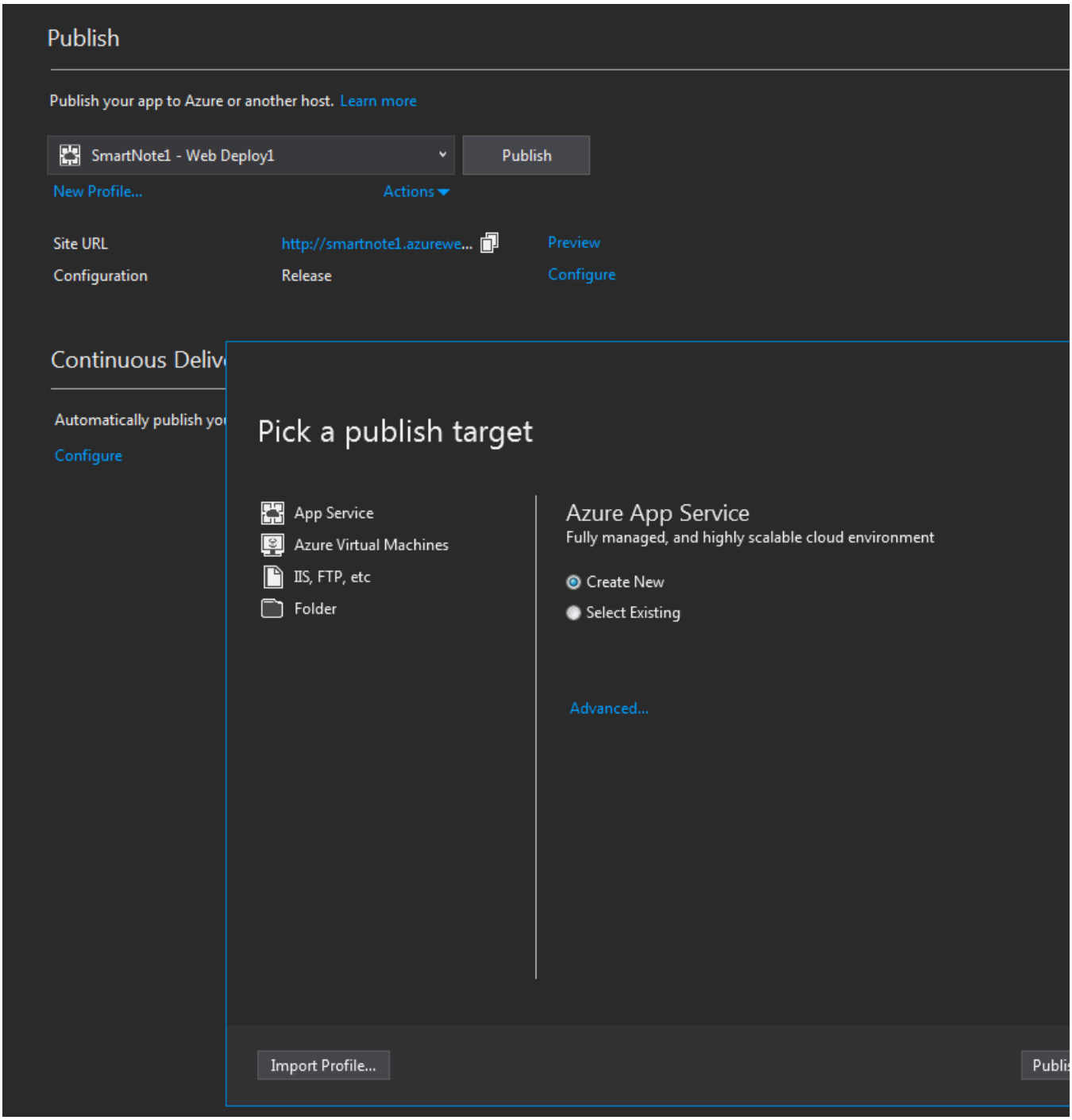

Figure 15. Download Profile

Step 5) It will publish but to connect the database you need to go to your SQL Database resource in your azure and copy your connection string.

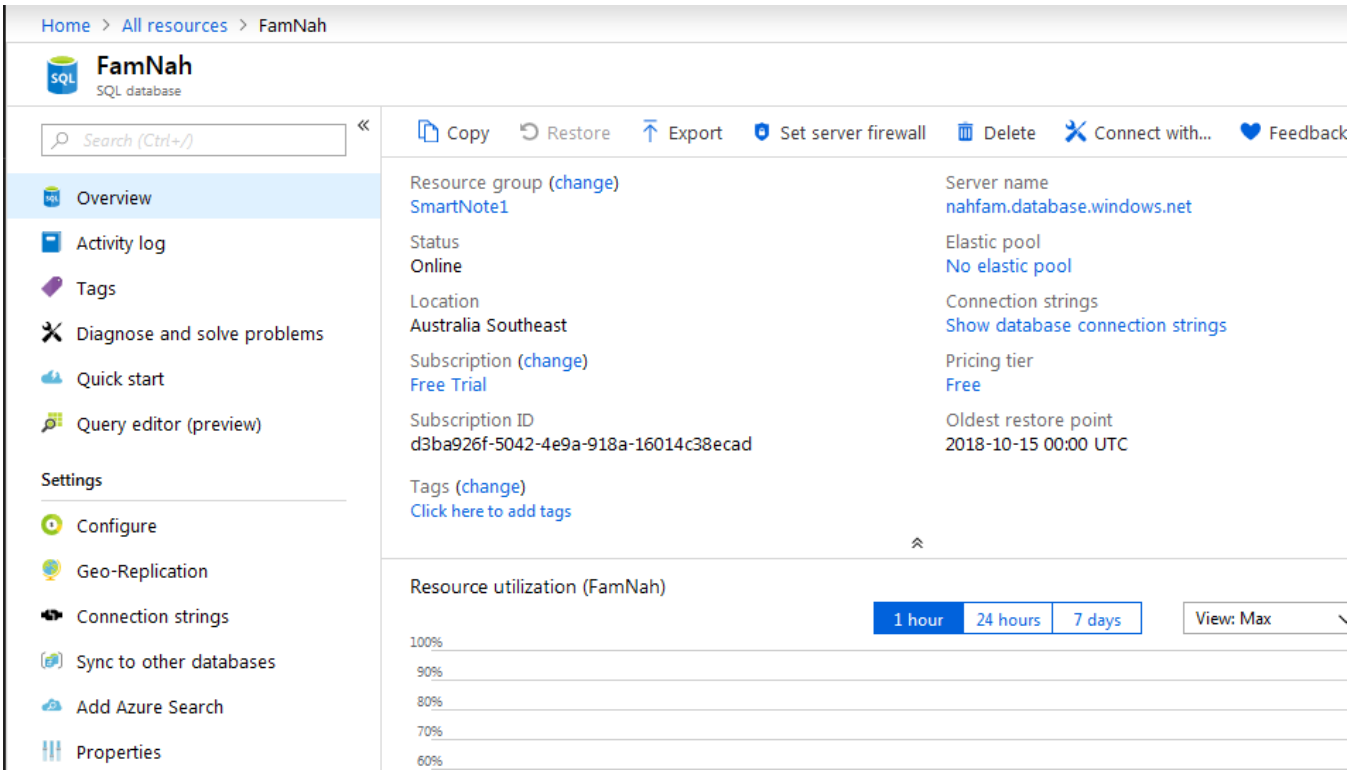

Figure 16. Copy Connection String

Step 6) With the copied connection string go to you publish profile > configure > settings and under databases paste the string, then click the 3dots and add your database username and password.

And finally re-publish and you're done!

6.0 - Appendices

Figure Number	Figure Name	Section
Figure 1	System Overview Design	2.1
Figure 2	Companion App Login Screen	3.1
Figure 3	Companion App Home Screen	3.2
Figure 4	Companion App Create Meeting Screen	3.3
Figure 5	Companion App Join Meeting Screen	3.4
Figure 6	Companion App Meeting Mode Screen	3.5
Figure 7	Companion App Past Meeting Screen	3.6
Figure 8	Companion App Upcoming Meeting Screen	3.7
Figure 9	Companion App Play Meeting Screen	3.8
Figure 10	Pi App Home Screen	3.9
Figure 11	Pi App Fast Create Screen	3.10
Figure 12	Pi App Scheduled Meeting Screen	3.11
Figure 13	Pi App Current Meeting Screen	3.12
Figure 14	Project Resources	5.0
Figure 15	App Service	5.0
Figure 16	Download Profile	5.0
Figure 17	Copy Connection String	5.0

Appendix 1. Table of Figures

Table Number	Table Name	Section
Table 1	App functionality	3.0
Table 2	Pi functionality	3.0

Appendix 2. Table of Tables

7.0 - Bibliography

- CSE3/5PRA & CSE3/5PRB Industry Project - 2018 Handbook

- Lecture materials:
  - Week 2 - Usability Teasting-2018
  - Week 8 - Software Documentation

- Australia and New Zealand Spatial Marketplace (ANZSM) - User Document Sample

- Wikipedia for:
  - .Net
  - Azure
  - Bootstrap
  - Class Diagram
  - CSS
  - ER
  - HTML
  - HTTP
  - JS
  - Raspberry Pi
  - React
  - Redux
  - SQL
  - Visual Studio
  - Wireframe

8.0 - Glossary

.NET

A software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large class library named Framework Class Library (FCL) and provides language interoperability (each language can use code written in other languages) across several programming languages.

Agile

Agile software development describes an approach to software development under which requirements and solutions evolve through the collaborative effort of self-organizing and cross-functional teams and their customer(s)/end user(s).

Application Programming Interface (API)

API is a set of subroutine definitions, protocols, and tools for building application software.

Architecture Diagram

An architectural model is a rich and rigorous diagram, created using available standards, in which the primary concern is to illustrate a specific set of tradeoffs inherent in the structure and design of a system.

Azure

Service created by Microsoft for building, testing, deploying, and managing applications and services through a global network of Microsoft-managed data centers.

Bootstrap

Bootstrap is a free and open-source front-end framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions.

Class Diagram

A class diagram in the Unified Modeling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects.

Cascading Style Sheets (CSS)

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language like HTML.

Entity Relationship (ER)

Describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types.

Hypertext Markup Language (HTML)

The standard markup language for creating web pages and web applications.

Hypertext Transfer Protocol (HTTP)

An application protocol for distributed, collaborative, and hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web.

Java Script (JS)

A common high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm.

Raspberry Pi

A series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries.

React

A JavaScript library for building user interfaces.

Redux

An open-source JavaScript library for managing application state. It is most commonly used with libraries such as React or Angular for building user interfaces.

Structured Query Language (SQL)

SQL is a domain-specific language used in programming and designed for managing data held in a relational database management system, or for stream processing in a relational data stream management system.

Visual Studio

An integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as web sites, web apps, web services and mobile apps

Wire Frame

Is a visual guide that represents the skeletal framework of a website. The wireframe depicts the page layout or arrangement of the website's content, including interface elements and navigational systems, and how they work together.

9.0 - Index

Space Index

Total number of pages: 87

0-9

Page: 1.0 - Introduction
Page: 1.0 Introduction
Page: 1.1 Audience Description
The document aims to inform the two main users of this software system - general staff (employees) and specialised staff (admins) - on how to install and use the software as well as update the system database for admin users. The regular staff will be the
Page: 1.1 System Overview
SmartNotes is a software solution that hopes to makes it easier for meeting attendees to record notes and tasks. The idea is built around a voice recording device to track meeting conversations. The meetings are recorded through a Raspberry Pi with a 360°
Page: 1.2 Applicability Statement
The companion App section on the system is designed to run on any device that has a browser; or in the case of smart phones it can also be launched natively through an app icon. The raspberry pi interface is designed to run the specific hardware device pr
Page: 1.2 Audience Description
The team has identified two primary users for this software system, but this may change throughout the development process. These users are general staff (employees) and specialised staff (admins). The regular staff will be the primary user base of this
Page: 1.3 Applicability Statement
The companion App section on the system is designed to run on any device that has a browser; or in the case of smart phones it can also be launched natively through an app icon. The raspberry pi interface is designed to run the specific hardware device pr
Page: 1.3 Purpose Statement
The main purpose of the SmartNotes software system is to provide employees with a simple, quick and intuitive means to create and review various meeting applications, such as notes, tasks, meeting audio etc. It's designed to remove hassle from meetings e
Page: 1.4 Document Usage Description
The User Document contains the following sections: Introduction Brief introduction of the system and its purpose and users. Introductory Kit System features and quick guide. SmartNote User Manual List of services the app provides and a description of each
Page: 1.4 Purpose Statement
The main purpose of the SmartNotes software system is to provide employees with a simple, quick and intuitive means to create and review various meeting applications, such as notes, tasks, meeting audio etc. It's designed to remove hassle from meetings e
Page: 1.5 - Conventions
The conventions used to create the User Document include headings, sizing, spacing, section separation, navigation to help identify the different parts contained within this document. The document was created on confluence using children for each of the
Page: 1.5 Document Usage Description
The System Maintenance Document contains the following sections: Introduction Brief introduction or overview of the projects system, purpose and its users and the System Maintenance Document itself. Software Design Scope Covers main functionality of the s
Page: 1.6 Conventions
The conventions used to create the System Maintenance Document include headings, sizing, spacing, section separation, navigation to help identify the different parts contained within this document. The document was created on confluence using children fo
Page: 1.7 Change Log

A

Version Date Changes made Author 1.0 May 30, 2018 Created document outline and all headings from 1.0 - 13.0 including all subsections. James Gherbaz 1.1 June 7, 2018 Updated sections: 2.1 Major Software Functions, 2.2 Major Design Constraints, 3.1 Existin

Page: 10.0 Reference Materials

CSE3/5PRA & CSE3/5PRB Industry Project - 2018 Handbook Lecture materials: Week 2 - Sprint Planning and JIRA 2018 Week 3 - User Stories, Size Estimation and Sprint Planning 2018 Week 5 - UI Prototyping 2018 Wikipedia for: .Net Azure Bootstrap Class Diagra

Page: 11.0 Index

Page: 12.0 Statement of Effort

Section Page Members contributed 1.1 System Overview Adam 1.2 Audience Description Adam 1.3 Applicability Statement Adam 1.4 Purpose Statement Adam 1.5 Document Usage Description Adam 1.6 Conventions Adam 1.7 Change Log James, Adam 2.1 Major Software Func

Page: 13.0 Inspect and Verified by (Quality Assurance)

Inspection and verification of this document has been performed by Adam Sanders in order to ensure the maximum quality of deliverable documents.

Page: 2.0 - Introductory Kit

Page: 2.0 Software Design Scope

Page: 2.1 Major Software Functions

General Users Create meeting and populate list of attendees Start, pause, resume and stop meetings Take timestamped notes during meeting Review past meetings and notes taken, take additional notes View upcoming meetings System Administrator Add and remove

Page: 2.1 SmartNotes Overview

SmartNotes is a software solution that hopes to makes it easier for meeting attendees to record notes and tasks. The idea is built around a voice recording device to track meeting conversations. The meetings are recorded through a Raspberry Pi with a 360°

Page: 2.2 Getting Started

This section gives a brief overview of the steps needed for the two different kinds of users to get started. It is a quick reference for the different actions different users can perform. There are two different apps that a user interacts with: a person

Page: 2.2 Major Design Constraints and other Requirements

Requirements User must be logged in to access all system functionality Constraints .NET framework for system backend React js for user companion app Raspberry pi audio recording device

Home page: 2018 - PRA - TEAM E

This is the home page for your documentation space within Confluence. Documentation spaces are great for keeping technical documentation organised and up to date. 1incompleteCustomise the home page - Click "Edit" to start editing your home page2incomplete

Page: 2018-03-07 Meeting 1

Date Attendees 17993324 18913190 18405364 18937919 18932508 Absentees n/a Goals Finalise groups preferred meeting times. Share contact details. Discussion items Time Item Who Description Discussion / Outcome 10min Meeting times James Find available ti

Page: 2018-03-14 Meeting 2

Date Attendees 17993324 18405364 18913190 18937919 18932508 Absentees N/A Goals Confirming what to discuss on for first meeting with supervisor. Discussion items Time Item Who Description Discussion / Outcome 5min Agenda All Discuss and confirm our agen

Page: 2018-03-15 Meeting 3

Date Attendees 17993324 18405364 18913190 18937919 18932508 Absentees N/A Goals Meet supervisor and orientate ourselves with general project. Discussion items Time Item Who Description Discussion / Outcome 5min Programming Skills All Discuss what we we

Page: 2018-03-17 Meeting 4

Date Attendees James Edward Gherbaz Adam Matthew Sanders Calvin Ng Cameron David Moon Jonathan Michael Fleming Absentees N/A Goals Decide on Project preferences. Discussion items Time Item Who Description Discussion / Outcome 30min project preference A

Page: 2018-03-20 Meeting 5

Date Attendees 17993324 18405364 18913190 18937919 Absentees 18932508 Goals Discuss SmartNotes Project Specifications. Discussion items Time Item Who Description Discussion / Outcome 10min specifications All analyse project description to determine spec

Page: 2018-03-28 Meeting 6

Date Attendees 17993324 18405364 18913190 18937919 18932508 Absentees N/A Goals Determine some quality questions for Q&A session with client Discussion items Time Who Description Discussion / Outcome 30mins All Discussed possible client questions Real-t

Page: 2018-04-09 Meeting 7

Date Attendees James Edward Gherbaz Adam Matthew Sanders Calvin Ng Cameron David Moon Jonathan Michael Fleming Absentees N/A Goals assign tasks to complete by next sprint Discussion items Time Item Who Description Discussion / Outcome 20mins review Q&A

Page: 2018-04-13 Meeting 8

Date Attendees 17993324 18405364 18913190 18937919 18932508 Absentees N/A Goals Inform/discuss with supervisor what we had completed in the sprint and what can be done and improved next sprint. Discussion items Time Item Who Description Discussion / Outc

Page: 3.0 - SmartNotes User Manual

This section gives the users a comprehensive view and guide for all of the screens that they may encounter in this system. There are 12 screens in total; the screens allow a user to complete all of the tasks outlined in the introductory kit. The followin

Page: 3.0 Reference Documents

Page: 3.1 Existing Software Documentation

The existing documentation consists of the system description and requirements document provided by Aerion Technologies.

Page: 3.1 User login

This is the first screen a user is met with when first loading up the companion app; it is a very simple screen which enables a user to login so that they can access their relevant meeting data. Figure 2 is the wire-frame of the login screen. Login.png F

Page: 3.10 Fast create (Impromptu meeting)

The impromptu meeting is for when a meeting is needed but the users have not scheduled a meeting previously. The screen waits until at least one person to join before moving onto the current meeting screen, begin recording audio, and allowing users to ta

Page: 3.11 Scheduled meeting

This is the screen in which users enter a planned meeting code. If the code is valid the meeting will begin and users can join the meeting and begin taking notes and Start meeting (PI).png Figure 12. Pi App Scheduled Meeting Screen There are four possib

Page: 3.12 Current meeting

The current meeting screen displays all of the current meeting information for the meeting that is in progress. The audio is recording at this point and users are able to join and make notes through their companion app. When the meeting is over as soon

Page: 3.2 Home page

This is the main screen of the companion app; from here navigation to all menus is possible. The persons user name is displayed across the top of the screen and their personalised list of upcoming meetings is shown. Only meetings in which the have enter

Page: 3.2 System Documentation

As this system is not being embedded into any existing system no system documents are applicable.

Page: 3.3 Create meeting

This section enables a user to plan an upcoming meeting. The meeting automatically generates a unique code, which then enables other users to access this meeting. For security reasons there is no way to publish/share this code in the app, users must dis

Page: 3.3 Vendor Documentation

Azure Microsoft Azure is a cloud computing service used for building, testing, deploying and managing applications and services. It provides the infrastructure for software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service

Page: 3.4 Join a meeting

In order to join a meeting two conditions must be met: The meeting code must exist (be valid) and the meeting must currently be in progress (from the pi app). If these conditions are met the meeting mode is enabled (that screen is shown in section 3.5)

Page: 3.4 Other Documentation

.NET A software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large class library named Framework Class Library (FCL) and provides language interoperability (each language can use code written in other languages)	
B	**C**
D	**E**
F	**G** Page: Getting started Here are some tips to get you started. You can edit this page to see how it works! 1. Create a page Click "Create" and select "Blank Page" to create your first page. New pages are created as children of the page you are currently viewing. 2. Add to your p
H	**I**
J	**K**
L	**M** Page: Making a template Give your authors a helping hand by using templates in your documentation space. You will need Space Administrator permissions to create templates. To create a template: Go to "Space Tools" in the sidebar, select "Content Tools" and create a new template. Page: Meeting Minutes Incomplete tasks from meetings All meeting notes
N	**O**
P Page: PRB Sprint 2 Review meeting Date Attendees 17993324 18405364 18913190 18937919 18932508 Absentees N/A Goals Inform/discuss with supervisor what we had completed in the sprint and what can be done and improved next sprint. Agenda Items Item Who Description Discussion / Outcome Confi	**Q**
R	**S** Page: System Maintenance Document (SMD) SmartNotes System Maintenance Document BY Team Eventually: Jonathan Fleming James Gherbaz Cameron Moon Calvin Ng Adam Sanders (Team Leader) FOR: College of Science Health and Engineering 2018 CSE3PRA/CSE3PRB Department of Computer Science at La Trobe Uni
T	**U** Page: User Document SmartNotes User Document BY Team Eventually: Jonathan Fleming James Gherbaz Cameron Moon Calvin Ng Adam Sanders (Team Leader) FOR: College of Science Health and Engineering 2018 CSE3PRA/CSE3PRB Department of Computer Science at La Trobe University UNDER
V	**W**
X	**Y**
Z	**!@#$**