Visitor Management System using Microservices Architecture

October 2019

By

William John Noel Mumby

Student number 201722044

Word count: 3284

1 CONTENTS

2 TABLE OF FIGURES

3 PROJECT BACKGROUND AND PURPOSE

3.1 BACKGROUND

Microservices is a buzz word at present in the distributed systems world but the core concepts and principles have been around for a long time in the form of Service Orientated Architecture (SOA). However, SOA had many fundamental flaws which stopped it from really taking off, microservices on the other hand have evolved from many of the lessons that were learnt by organisations who did SOA badly. In the words of Sam Newman "The microservices approach has emerged from real world use, taking our better understanding of systems and architecture to do SOA well." (Newman Sam, 2015:8)

Currently the de facto standard way to make a backend service is within one giant code base which contains all the code for the entire system and the data is often stored in one relational database; this is known as the monolithic approach. (Dragoni et al., 2017). There is nothing wrong with this approach and in a lot of cases were the application is small and may not require the benefits that microservices offer then this design makes perfect sense (Gankiewicz Piotr, n.d.). However, in some cases when companies are operating at huge scale such as Amazon and Netflix this is not maintainable and soon gets out of control; beginning to cost a lot of money to scale to user demands (Newman Sam, 2015).

This is where the microservices architecture comes in, Martin Fowler describes the microservice architecture as an "approach to developing a single application as a suite of small services each running in its process." (Fowler, 2019). This is a good starting point as the main difference between a monolithic approach and a microservice based one is that each microservice is its own application whereas a monolithic code base may be separated into components and systems, but these are imaginary boundaries whereas microservices are physical ones. This allows for proper separation of concerns modelled around the domain in which the application is being built.

3.2 Aims and Objectives

The aim of the project is to create a visitor management system while investigating the microservice architecture through these ends. This seeks to streamline the management of visitors and staff for a business across multiple sites. It aims to save time for receptionists and employees who use the system. This system is intended to be managed by a company who can distribute this service to other companies using a system management portal. The user side of the system will comprise of two web front ends and a portable app which consumes the set of microservices providing different user access levels and experiences. The system is explained in more detail in Appendix D – Project Description. The objectives for the project are defined below:

Primary Objectives

These objectives are the main goals of the project and are aimed to be completed in full.

# Objective 1: Create a backend system which provides the capability defined in Appendix A – Requirements for *features 1,2 and 3*.

# Objective 2: Create a user interface in the form of a web application which harnesses the capability provided by the backend system to supply *feature 1 system admin portal (*Appendix A – Requirements*).*

# Objective 3: Create a user interface in the form of a web application which harnesses the capability provided by the backend system to supply *feature 2 business portal (*Appendix A – Requirements*).*

# Objective 4: Create a user interface in the form of a web application which harnesses the capability provided by the backend system to supply *feature 3 greeting system (*Appendix A – Requirements*).*

# Objective 5: Utilise authorisation/authentication in line with the JWT specification RFC 7519 (Bradley et al., 2015).

Secondary Objectives

These are objectives, that if time allows, will be completed to enhance the final product.

# Objective 6: Extend the capability of the backend system to be tailored towards a mobile application to provide the capability defined in *feature 4 business member app (*Appendix A – Requirements*).*

#Objective 7: Create a user interface in the form of a web application which harnesses the capability provided by the backend system to supply *feature 4 business member app (*Appendix A – Requirements*).*

3.3 Scope and deliverables

The project will deliver a fully functioning backend which is consumed by multiple web applications as stated in the objectives. In addition to this, a Universal Windows Platform (UWP) application will be created which performs the sign in for visitors.

The scope of this project is to complete the objectives detailed in the section above to a point in which the system meets the requirements. This shall include unit testing in areas of the system that are beneficial. In addition to this acceptance testing will also be conducted against each user story defined in the requirements (Appendix A). This will allow the project to be assessed against its objectives as these tests will prove that each piece of user functionality has been met to the customers desire.

The scope will not include deployment of the system by the end of this project. In addition to this integration testing will not be performed in this project due to time constraints. There will also be no formal documentation for the code base. The customer has also expressed the wishes for many more features such as badge printing. These have been declared out of scope for this project.

The use of the microservice architecture will be shown and assessed through the design of the system and in demos where the different services can be clearly identified under the business domain. It should also show a high level of resilience through demonstration which aims to show the system operating with one or more services disabled.

3.4 Constraints and assumptions

Alternate working commitments could impact the project as the author also works at BAE Systems and will return there in the holiday periods. This may require time to be balanced between this project and work for the company; reducing the time in which the author can spend on the project in these periods.

The use of libraries and open source frameworks is inevitable in such a project. Anything that has been made open source more than likely comes attached with a license, it is important that the project is compliant with these licenses. In some cases, these licenses could impact the project in terms of how external open source libraries can be used.

One assumption is that the system will be designed in a way which will allow for easy expansion and modification in the future. The use of microservices promotes this but adds complexity compared to a traditional single process application. A big assumption is that the system will be built upon in the future which justifies this specific architecture. This is justified as it promotes best practices and will give the company the ability to run the system at scale in the scenario that they end up with thousands of users in the future.

There have also been some assumptions made in terms of the client’s needs from the system this is mainly related to the accepted quality of the final product. There has been a lot of ideas in which the project can be improved in the future by the customer. In addition to this, there is no performance requirements placed on the project by the customer as they do not know at what scale it will need to operate. Furthermore, the security level has been assumed to be high as it stores customers data.

4 PROJECT RATIONALE AND OPERATION

4.1 PROJECT BENEFITS

The benefits of the system will be to reduce costs attached to visitor management. Last updated on September 13th, 2019 the average salary per year for a receptionist is £17,593 (PayScale, n.d.). In addition to this it also benefits a business's security knowing who has been and will be on site. This also ensures fire safety by providing a digital up to date fire list. It is stated in government regulations the individual/individuals in charge of a business are responsible for fire safety for both visitors and staff (GOV UK, n.d.).

In addition to the final product the background section of the final report aims to provide material which can be used as reference for individuals and organisations who are interested or considering the microservices architecture for their systems. It aims to provide guidance in the selection process for when to and not to use microservices. Furthermore, it aims to present some of the different options that can be taken from within the architecture highlighting their benefits and drawbacks; allowing the reader to make informed decisions when implementing this architecture.

4.2 PROJECT OPERATION

The project will be operated with the main aim to provide value to the customer of the product on a frequent basis, fitting in with the agile values and principles. The project aims to operate in an agile manner targeting chunks of work over 2-3 week periods which provides some visible benefit to the customer.

This will entail of consistent reviews of the backlog aiming to keep it prioritised and delivering milestones which are set, on time. The way in which the software is developed will also follow the agile methodology which aims to provide vertical slices of functionality from the ground all the way up to the user interface. This will be supported by the microservice architecture which directly compliments this. Furthermore, the code will be written in way which aims to be compliant with the SOLID coding principles which make code which is not heavily affected by change to requirements (Hall, 2014). Unit testing will also be conducted in relevant areas of the code to help maintain quality and a design in line with the SOLID principles.

The success of the chosen method in which to operate the project will be assessed by monitoring the ability of the project to meet the detailed milestones on time and to a standard which meets all its acceptance criteria.

4.3 Options

The list below highlights some of the technologies, frameworks, languages and Integrated Development Environments (IDE) available to the project.

IDE's:
- Visual Studio
- Visual Studio Code
- Rider
- Atom

Languages:
- C#
- F#
- JavaScript
- Java
- Go

Server-Side Frameworks:
- ASP.NET
- ASP.NET CORE
- Node.js
- Spring

Client-Side Frameworks:
- Angular
- React
- Vue.js
- Razor
- Blazor

Data Storage:
- Mongo Db
- Cosmos Db
- SQL Server
- MySQL

Mobile Platforms:
- Xamarin
- Cordova
- React Native
- Android
- IOS
- UWP

Design Patterns:
- SOLID
- CQRS
- REST
- MVVM
- BaaS

Communication Mechanisms
- RabbitMQ
- Azure Service Bus
- N Service Bus
- HTTP
- RPC

See Appendix C – Acronym Dictionary for acronym definitions.

All applications will all make use of SOLID design principles as this promotes unit testing and allows code to adapt to change. (Hall, 2014)

The microservice backend will make use of a combination of Visual Studio and Visual Studio Code for its IDE as it best supports the chosen language (C#) and framework (ASP.NET CORE). This decision has been made due to the unit testing frameworks available, the community support and the cross-platform capabilities of ASP.NET CORE which opens the hosting options in the future. In terms of data storage, it will use Mongo Db as its document-based storage allows for flexibility in domain models and support for scalability (Mongo, n.d.). This will also make use of SOLID to improve code adaptability. CQRS will also be used to provide a clear separation of concerns between the read and write side of each service (Newman Sam, 2015). The inter-service communication will utilise RabbitMQ as it provides a client library for C#, is asynchronous (non-blocking) and is open source (RabbitMQ, n.d.). Client communication will use HTTP as it provides documentable mechanism for data transfer and is also synchronous which makes response handling easier for clients.

The front-end web applications that are to utilise the backend services in the form of the *Business Portal* and the *System Admin Portal* will again use a combination of Visual Studio and Visual Studio

Code as the language and framework (C# and Blazor) are supported very well in both editors. Blazor has been chosen for its component-based design which allows of sections of user interface to be re-used as well as its easily changed hosting model whether it is server-side or client side. This will allow for scaling of the application to be managed.(Microsoft, n.d.a)

The *Greeting System* is to be used on a form of tablet therefore the UWP platform has been selected as the customer wishes to keep the cost of tablets low. In addition, it supports the MVVM pattern which allows for an application to split up which makes unit testing a lot easier (Microsoft, n.d.b). Due to the platform chosen C# and Visual Studio will be used to develop the application.

The choices that are made will be assessed at the end of the project using a retrospective analysis aiming to answer three simple questions, what went well, what didn't go so well and what would be done differently. These will be recorded for any future projects that should be carried out so the lessons learned can be filtered through to help promote continuous improvement, one of the core takeaways from agile.

4.4 Risk analysis

The following risk analysis will describe a risk, how it can be mitigated, the likelihood (L), the severity (S) and the impact. The likelihood and the severity will be ranked 1-5, 1 being the least likely and 5 being the most. These will be used to calculate the impact using (*L x S = I)*.

Risk	**Mitigation**	**Recovery**	**L**	**S**	**I**
Hard disk failure	Use cloud services such as one drive for documentation and source control providers such as GitHub for code.	Backup work onto a memory stick periodically.	3	2	6
Requirement Ambiguity	Agree defined acceptance criteria with customer to allow a requirement to be marked as complete.	Rework acceptance criteria to be defined and concise.	3	3	9
Design lacks Flexibility	Make use of agile coding practices such as SOLID to allow code to be changed with large impacts.	Highlight areas and make the necessary design changes.	2	2	4
Short term illness	N/A	Make up extra work on the weekends.	5	1	5
Long term illness	N/A	N/A	2	5	10
Supervisor absence	Have a detailed plan to allow for self-working.	Contact second marker or module leader for extra support.	2	2	4
Customer Rejection	Spend time upfront making sure that the requirements and acceptance criteria reflect the customers wishes.	Address customer reasons for rejection.	2	3	6
Lack of time	Make use of the plan and the agile tools provided by GitHub to manage tasks.	Work extra across weekends or reduce project scope.	2	3	6
Lack of task management	Make use of GitHub labels to clearly identify tasks related to certain milestones and aims.	Stop and re-categorize tasks for remaining work.	2	1	2
Bad library selection	Wrap libraries so they can be easily changed.	Change the library for a better alternative.	2	2	4

4.5 Resources required

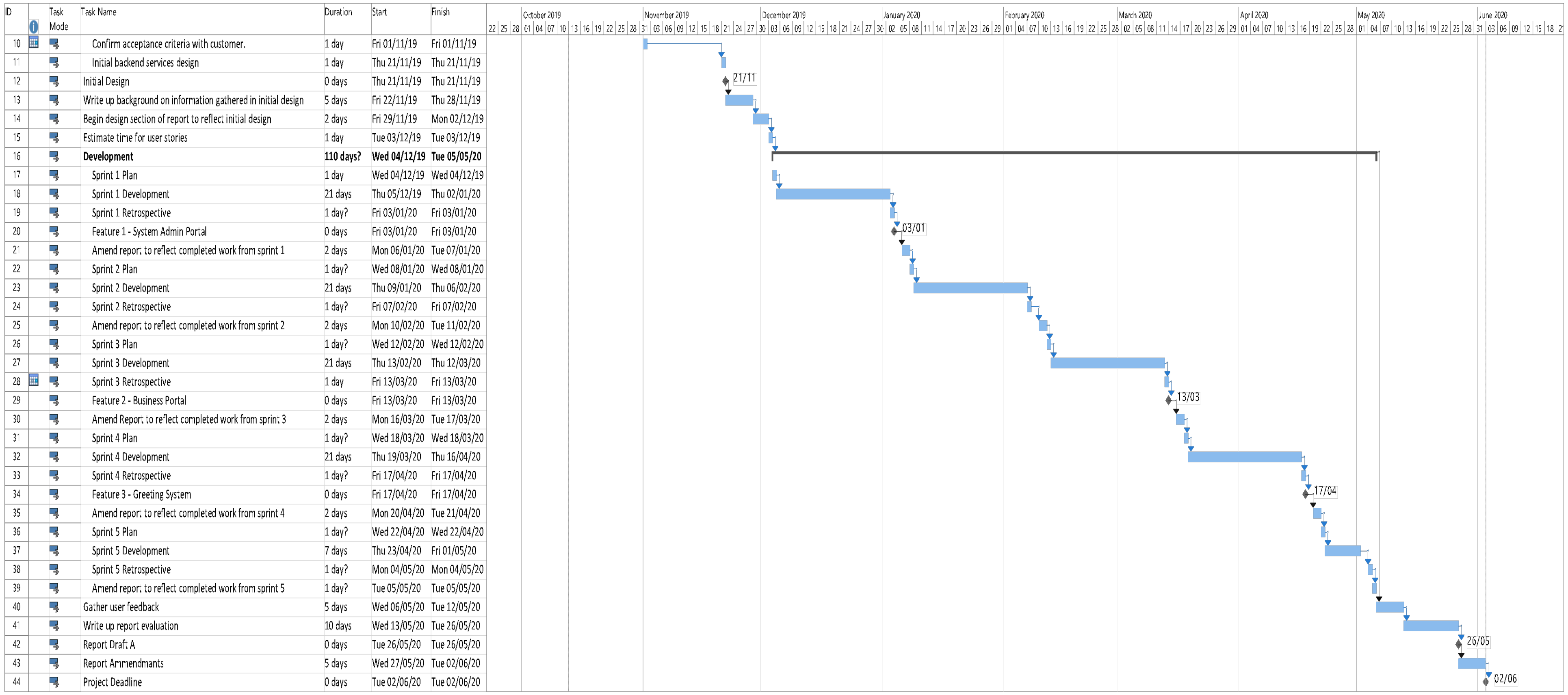
There are no non-standard resources required for this project.

5 Project Methodology and Outcomes

5.1 Task & Milestones Breakdown

Task	**Days**
Research and identify language and frameworks to be used.	2
Research and identify inter-service communication method.	2
Research and identify host platform to run services.	2
Research and identify a data storage solution.	2
Implement library for common inter-service communication.	3
Implement library for common data access across services.	1
Research and identify auth method for security.	4
Implement or identify a library to be used for auth.	2
Confirm acceptance criteria with customer.	1
Initial backend services design	2
[MILESTONE] – Initial Design	21
Write up background on information gathered in initial design.	5
Begin design section of report to reflect initial design	2
Estimate time for user stories.	1
Sprint 1 Plan	1
Sprint 1 Development	21
Sprint 1 Retrospective	1
[MILESTONE] – Feature 1: System Admin Portal	0
Amend report to reflect completed work from sprint 1.	2
Sprint 2 Plan	1
Sprint 2 Development	21
Sprint 2 Retrospective	1
Amend report to reflect completed work from sprint 2.	2
Sprint 3 Plan	1
Sprint 3 Development	21
Sprint 3 Retrospective	1
[MILESTONE] – Feature 2: Business Portal	0
Amend report to reflect completed work from sprint 3.	2
Sprint 4 Plan	1
Sprint 4 Development	21
Sprint 4 Retrospective	1
[MILESTONE] – Feature 3: Greeting System	0
Amend report to reflect completed work from sprint 4.	2
Sprint 5 Plan	1
Sprint 5 Development	7
Sprint 5 Retrospective	1
Amend report to reflect completed work from sprint 2.	2
Gather User Feedback	5
Write up report evaluation	10
[MILESTONE] – Report Draft A	0
Report amendments	5
[MILESTONE] – Project Deadline	0
Total	181

5.2 Schedule Gantt Chart



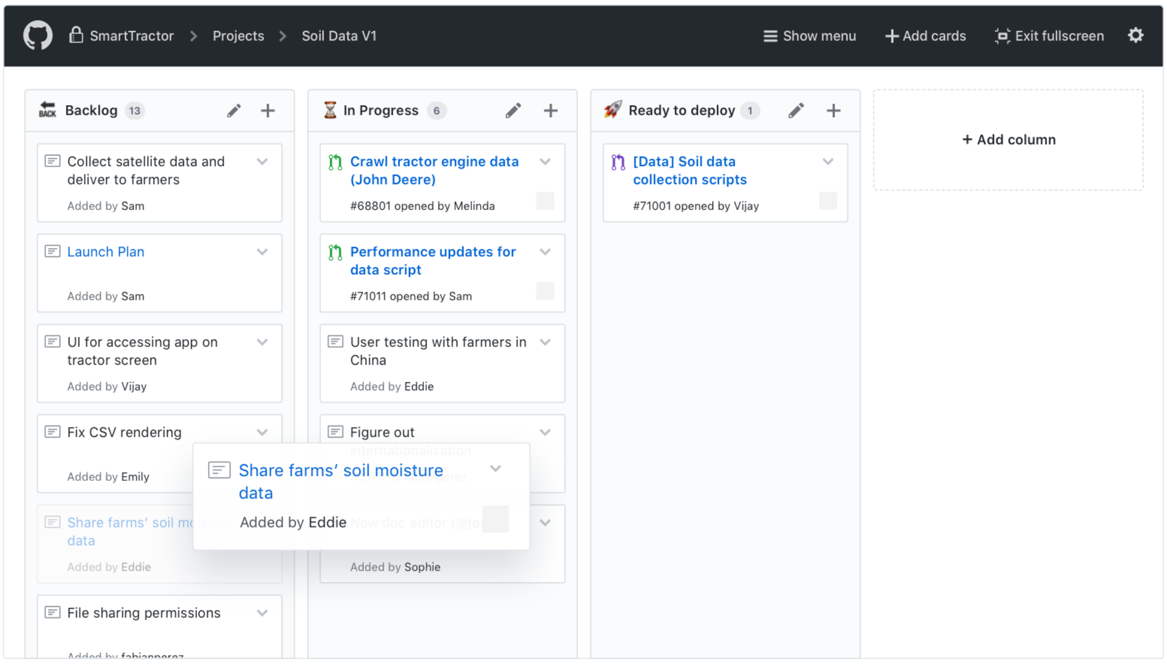
ID	Task Mode	Task Name	Duration	Start	Finish
10		Confirm acceptance criteria with customer.	1 day	Fri 01/11/19	Fri 01/11/19
11		Initial backend services design	1 day	Thu 21/11/19	Thu 21/11/19
12		Initial Design	0 days	Thu 21/11/19	Thu 21/11/19
13		Write up background on information gathered in initial design	5 days	Fri 22/11/19	Thu 28/11/19
14		Begin design section of report to reflect initial design	2 days	Fri 29/11/19	Mon 02/12/19
15		Estimate time for user stories	1 day	Tue 03/12/19	Tue 03/12/19
16		**Development**	**110 days?**	**Wed 04/12/19**	**Tue 05/05/20**
17		Sprint 1 Plan	1 day	Wed 04/12/19	Wed 04/12/19
18		Sprint 1 Development	21 days	Thu 05/12/19	Thu 02/01/20
19		Sprint 1 Retrospective	1 day?	Fri 03/01/20	Fri 03/01/20
20		Feature 1 - System Admin Portal	0 days	Fri 03/01/20	Fri 03/01/20
21		Amend report to reflect completed work from sprint 1	2 days	Mon 06/01/20	Tue 07/01/20
22		Sprint 2 Plan	1 day?	Wed 08/01/20	Wed 08/01/20
23		Sprint 2 Development	21 days	Thu 09/01/20	Thu 06/02/20
24		Sprint 2 Retrospective	1 day?	Fri 07/02/20	Fri 07/02/20
25		Amend report to reflect completed work from sprint 2	2 days	Mon 10/02/20	Tue 11/02/20
26		Sprint 3 Plan	1 day?	Wed 12/02/20	Wed 12/02/20
27		Sprint 3 Development	21 days	Thu 13/02/20	Thu 12/03/20
28		Sprint 3 Retrospective	1 day	Fri 13/03/20	Fri 13/03/20
29		Feature 2 - Business Portal	0 days	Fri 13/03/20	Fri 13/03/20
30		Amend Report to reflect completed work from sprint 3	2 days	Mon 16/03/20	Tue 17/03/20
31		Sprint 4 Plan	1 day?	Wed 18/03/20	Wed 18/03/20
32		Sprint 4 Development	21 days	Thu 19/03/20	Thu 16/04/20
33		Sprint 4 Retrospective	1 day?	Fri 17/04/20	Fri 17/04/20
34		Feature 3 - Greeting System	0 days	Fri 17/04/20	Fri 17/04/20
35		Amend report to reflect completed work from sprint 4	2 days	Mon 20/04/20	Tue 21/04/20
36		Sprint 5 Plan	1 day?	Wed 22/04/20	Wed 22/04/20
37		Sprint 5 Development	7 days	Thu 23/04/20	Fri 01/05/20
38		Sprint 5 Retrospective	1 day?	Mon 04/05/20	Mon 04/05/20
39		Amend report to reflect completed work from sprint 5	1 day?	Tue 05/05/20	Tue 05/05/20
40		Gather user feedback	5 days	Wed 06/05/20	Tue 12/05/20
41		Write up report evaluation	10 days	Wed 13/05/20	Tue 26/05/20
42		Report Draft A	0 days	Tue 26/05/20	Tue 26/05/20
43		Report Ammendmants	5 days	Wed 27/05/20	Tue 02/06/20
44		Project Deadline	0 days	Tue 02/06/20	Tue 02/06/20

Figure 1 Project Gantt Chart

5.3 Project Control

The project shall be managed day to day using the GitHub project management tools. This allows an Agile workflow which can be directly linked to source code using Git.

This provides a way to create tasks in the form of issues, categorize these and then group them using milestones. This allows for an agile workflow which can be visualised by making using of project boards where tasks and milestones can be assigned to a specific project. This is shown in Figure 2 Project Boards in GitHub (GitHub, n.d.b).



Backlog 13

- Collect satellite data and deliver to farmers — Added by Sam
- Launch Plan — Added by Sam
- UI for accessing app on tractor screen — Added by Vijay
- Fix CSV rendering — Added by Emily
- Share farms' soil moisture data — Added by Eddie
- File sharing permissions — Added by fabianperez

In Progress 6

- Crawl tractor engine data (John Deere) — #68801 opened by Melinda
- Performance updates for data script — #71011 opened by Sam
- User testing with farmers in China — Added by Eddie
- Figure out internationalization
- New doc editor (@jo
- Added by Sophie

Ready to deploy 1

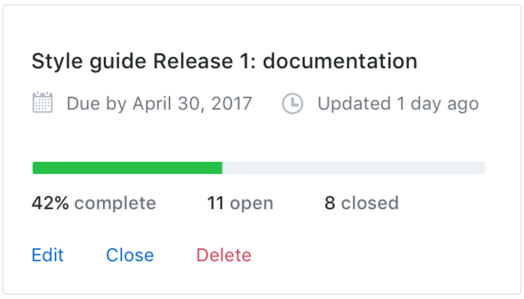
- [Data] Soil data collection scripts — #71001 opened by Vijay

Share farms' soil moisture data — Added by Eddie

Figure 2 Project Boards in GitHub (GitHub, n.d.b)

Tasks which are represented as issues in GitHub can be given labels which have been predefined for this project and can be found in Appendix B – Task Categorization.

The performance will be monitored using milestones which are defined in the project plan and shown in the Gantt chart. This is shown in Figure 3 Milestone Progress in GitHub (GitHub, n.d.a).



Style guide Release 1: documentation

Due by April 30, 2017   Updated 1 day ago

42% complete   11 open   8 closed

Figure 3 Milestone Progress in GitHub (GitHub, n.d.a)

The success of the tool will be reviewed at the end of the project by the author to provide feedback in whether this tool helped, hindered or improved the performance of the project.

6 EVALUATION

The plan which is detailed in this document is based off the limited discussion with the customer in which the product is being created for shown by the specification Appendix A – Requirements. However, with the meetings to come with the consumer there will be likely change required to these or more expression inside of the acceptance criteria required to capture the full specification of the system. This may add or reduce the complexity and amount of work to complete the project.

The plan is a solid starting point but will require to be regularly revisited to ensure its effectiveness. This is complemented by the selected Agile methodology which is built around changing customer expectations and needs (Hall, 2014). This should allow the project to absorb any change and filter this into the initial plan.

7 Appendix A – Requirements

The requirements below are expressed as a set of user stories with a brief description of the user roles to help distinguish their access to the system.

User Roles

System Administrator

A system administrator will be in total control of the system, they will use the system to set-up new business as well as monitoring and updating existing businesses within the system.

Business Administrator

A business administrator will be in control of a single business which will be created by a *System Administrator* this will include employee and visitor management for the full business. A business administrator will also have all the capabilities of a *Business Member.*

Business Member

A business member will be able to use the sign in/out service as well as monitoring their own records alongside the records of their specific visitors.

Visitor

A visitor will be able to sign in and out of a business site upon entry and exit providing relevant information.

System Features

The following sections will breakdown the system into core features and then into the features uses from each user prospective. These will be written as user stories telling the usage of the system from the users prospective.

# Feature 1 System Admin Portal

The system admin portal shall only be accessible by a *System Administrator* so there are only one set of stories.

*As **a system admin** I want the system to be run locally on site so that its access is limited to people on our network.*

*As a **system admi**n I want the system to require a login so that the system can only be accessed by people with the correct credentials.*

*As **a system admin** I want to be able to view all the businesses that are currently using the system so I can keep track of all our customers.*

*As a **system admin** I want to be able to view the information attached to each business so that I can access their information in one central place.*

*As a **system admin** I want to be able to manage sites that a business has so that I can added and kept up to date.*

*As a **system admin** I want to be able to manage business administrator users for a specific business so that they can be added and kept up to date.*

As a system admin I want to be able to add a new business to the system so that we can get new businesses setup using the system.

# Feature 2 Business Portal

The business portal will be accessed by two different users the *Business Administrator* and a *Business Member*. The requirements for each user are broken down in the following user stories. A business admin will also have all the capabilities of a *Business Member* when accessing the portal.

Business Admin

*As a **business admin** I want to be able to add user accounts for my employees so that all my employees can access the system.*

*As a **business admin** I want to be able to manage user accounts for my employees so the system will always reflect my current workforce.*

*As a **business admin** I want to be able to view all the access records for all employees so that I can check the times they are working.*

*As a **business admin** I want to be able to view all the access records for all visitors for each site so that I can check the visitors who have been on each premises.*

*As a **business admin** I want to be able to define the information I wish to gather about a visitor so that I can capture the information specific to my business.*

Business Member

*As a **business member** I want to access the portal using my credentials*

*As a **business member** I want to be able to view all my personal access records to all sites so I can make sure the system is correct.*

*As a **business member** I want to be able to view all my visitors access records to all sites so I can refer to the information in the future.*

*As a **business member** I want to be able to view all the information collected for each of my visitors so that I can use the information later.*

*As a **business member** I want to be able to view all Business Members signed in at all sites so I can see who is available.*

*As a **business member** I want to be able to view all Business Members working off-site so I can assess the best method of contact.*

*As a **business member** I want access to a fire list so that I can start checking people off in the case of an emergency at the muster point.*

# Feature 3 Greeting System

The greeting system will be again used by two different users a *Business Member* and a *Visitor*. The requirements for each user will be broken down in the following user stories. In this system a *Business Member* will have access as a *Visitor*.

Business Member

*As a **business member** I want to be able to setup the system using our unique business credentials so that the system can be personalised for my business.*

*As a **business member** I want to be able sign in and out using unique credentials to me so that I can identify myself to the system.*

*As a **business member** I want to be notified when one of my visitors arrives so that I can go and greet them.*

Visitor

*As a **visitor** I want to be able to see the current time when using the system so I can make sure I am on running within my schedule for the day.*

*As a **visitor** I want to be able to identify the business when using the system so I can make sure I am using the correct system for the company that I am visiting.*

*As a **visitor** I want to be able to easily sign in by providing my information so that I can notify my host that I have arrived.*

*As a **visitor** I want to be able to opt in or out of my picture taken so I do not always have to have my picture taken.*

*As a **visitor** I want to be able to sign out so I can let my host know that I have left the premises.*

*As a **visitor** I want to be able to quickly enter my information using a touchscreen, so I do not have to mess around with any peripheral devices.*

# Feature 4 Business Member App

The mobile application will only be accessible will only be accessible to a *Business Member* so there will only be one set of stories.

*As a **business member** I want to be able to access the sign in system with credentials unique to me and my business so that I can keep my data secure.*

*As a **business member** I want to be able to sign in and out always tagging my location so that I can do this off site showing where I am.*

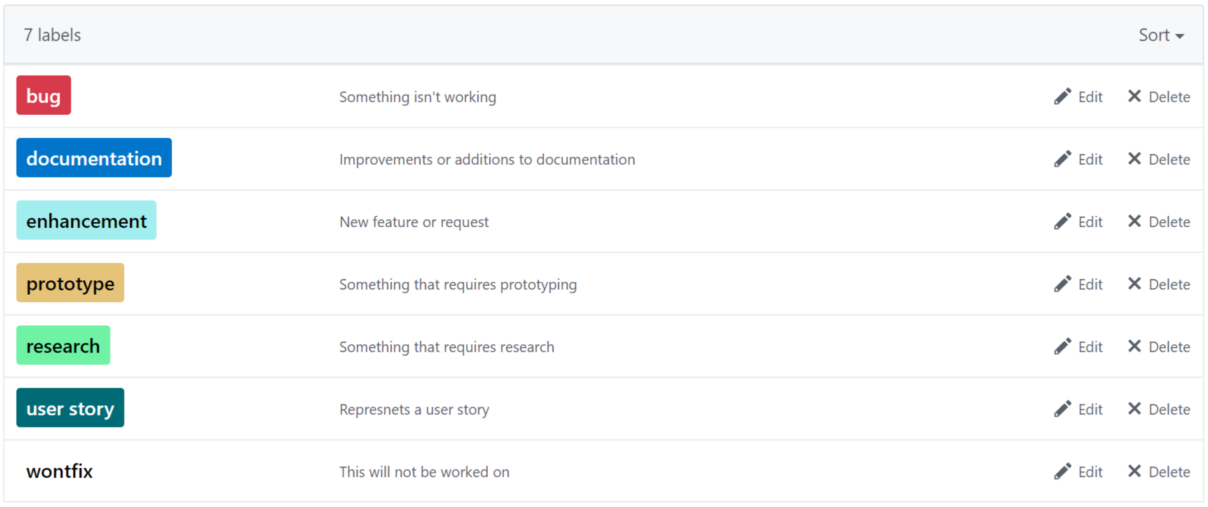
*As a **business member** I want to be able to view all my visitors so that I can keep track of them quickly from my mobile device.*

*As a **business member** I want to able to view the fire list for all sites so that I can access this quickly from my phone in the case of emergency.*

*As a **business member** I want to be able to view all employees currently on site so that I can use the best method of communication to contact them.*

8 APPENDIX B – TASK CATEGORIZATION

The table below shows how tasks can be filtered and categorized using the GitHub project management tool.



7 labels		Sort ▾
bug	Something isn't working	Edit ✕ Delete
documentation	Improvements or additions to documentation	Edit ✕ Delete
enhancement	New feature or request	Edit ✕ Delete
prototype	Something that requires prototyping	Edit ✕ Delete
research	Something that requires research	Edit ✕ Delete
user story	Represnets a user story	Edit ✕ Delete
wontfix	This will not be worked on	Edit ✕ Delete

9 APPENDIX C – ACRONYM DICTIONARY

Acronym	**Description**
MVVM	Model View View-Model
REST	Representational State Transfer
CQRS	Command Query Responsibility Segregation
SOLID	**S**ingle Responsibility Principle **O**pen Closed Principle **L**iskov's Substation Principle **I**nterface Segregation Principle **D**ependency Injection
BaaS	Backend as a Service
UWP	Universal Windows Platform
TFS	Team Foundation Services
HTTP	Hyper Text Transfer Protocol
RPC	Remote Procedure Call

10 Appendix D – Project Description

This project will be a web-based system that will allow a business to keep track of visitors and staff accessing multiple sites. This will allow a visitor to use a tablet at reception to fill in their details. These details will be customised by the business so they can collect the information specific to their business. Employees will also be able to use this facility to sign in and out of the site they can monitor these records as well as their own visitors in a portal which can be accessed on the internet. The portal will also have a business admin role who can monitor all visitors and staff as well as their own records. This will also be accessed from the internet.

The businesses that are created will be managed by the business who runs the full system. This will be done using a system admin portal. This will allow new businesses to be added to the system and then given credentials to access. This will allow the company who is selling the system to monitor and track all the businesses who are using the system as well as removing them.

The full project will be written using a backend system which is comprised of small services which follow the microservices architecture. This will allow for multiple lightweight clients to be created in order to utilise the services provided by the system for different use cases.

11 References

Bradley, J., Sakimura, N. & Jones, M. (2015) *JSON web token (JWT).* Available online: https://tools.ietf.org/html/rfc7519 [Accessed Oct 16, 2019].

Dragoni, N., Giallorenzo, S., Lafuente, A. L., Mazzara, M., Montesi, F., Mustafin, R. & Safina, L. (2017) Microservices: Yesterday, today, and tomorrow. In Manuel Mazzara and Bertrand Meyer (eds) *Present and ulterior software engineering.* Cham: Springer International Publishing, 195-216.

Fowler, M. (2019) *Microservices guide.* Available online: https://martinfowler.com/microservices/ [Accessed Oct 9, 2019].

Gankiewicz Piotr *.NET core microservices – theory.* Available online: https://piotrgankiewicz.com/2018/07/09/net-core-microservices-theory-dshop-solution-structure/ [Accessed Oct 15, 2019].

GitHub (n.da) *GitHub milestone.* Available online: https://github.githubassets.com/images/modules/site/product-illo/img-projects-milestones.png [Accessed 16 Oct, 2019].

GitHub (n.db) *GitHub project board.* Available online: https://github.githubassets.com/images/modules/site/product-illo/img-project-full.png [Accessed 16 Oct, 2019].

GOV UK (n.d.) *Fire safety in the workplace.* Available online: https://www.gov.uk/workplace-fire-safety-your-responsibilities [Accessed Oct 16, 2019].

Hall, G. (2014) *Adaptive code via C#: Agile coding with design patterns and SOLID principles*, 1st edition Microsoft Press.

Microsoft (n.d.a) *ASP.NET core blazor hosting models.* Available online: https://docs.microsoft.com/en-us/aspnet/core/blazor/hosting-models [Accessed Oct 16, 2019].

Microsoft (n.d.b) *Data binding and MVVM - windows UWP applications.* Available online: https://docs.microsoft.com/en-us/windows/uwp/data-binding/data-binding-and-mvvm [Accessed Oct 16, 2019].

Mongo *Advantages of NoSQL.* Available online: https://www.mongodb.com/scale/advantages-of-nosql [Accessed Oct 16, 2019].

Newman Sam (2015) *Building microservices*, 1st edition O'Reilly.

PayScale *Pay scale.* Available online: https://www.payscale.com/research/UK/Job=Front_Desk_Receptionist/Salary [Accessed Oct 16, 2019].

RabbitMQ (n.d.) *Features - rabbit MQ.* Available online: https://www.rabbitmq.com/features.html [Accessed 16 Oct, 2019].