TO

Abstract

[Draw your reader in with an engaging abstract. It is typically a short summary of the document.   
When you’re ready to add your content, just click here and start typing.]

Course: COM547 – Computing Systems

Supervisor: Dr Colin Turner

Date: 22/04/2018

Project Final Report

Student: B00348751 – Paul Connolly

# Acknowledgements

Contents

[Acknowledgements ii](#_Toc511578182)

[1 Introduction 1](#_Toc511578183)

[1.1 Background 1](#_Toc511578184)

[1.2 Project Aims 5](#_Toc511578185)

[1.3 Project Objectives 5](#_Toc511578186)

[1.4 Project Activities 5](#_Toc511578187)

[1.5 Outline of Dissertation Structure 6](#_Toc511578188)

[2 Chatbots in Use Today 8](#_Toc511578189)

[2.1 What is a chatbot? 8](#_Toc511578190)

[2.2 Review of Chatbots 8](#_Toc511578191)

[2.2.1 Marvel 8](#_Toc511578192)

[2.2.2 RoofAi 8](#_Toc511578193)

[2.2.3 Lemonade 11](#_Toc511578194)

[3 Technical Background 13](#_Toc511578195)

[3.1 Frameworks 13](#_Toc511578196)

[3.1.1 Microsoft Bot Framework 13](#_Toc511578197)

[3.1.2 Wit.ai 14](#_Toc511578198)

[3.1.3 DialogFlow 15](#_Toc511578199)

[4 Development Lifecycle 16](#_Toc511578200)

[4.1 Software Development Process Models Considered 16](#_Toc511578201)

[4.1.1 Waterfall Model 16](#_Toc511578202)

[4.1.2 Spiral Model 17](#_Toc511578203)

[4.1.3 Agile Model 19](#_Toc511578204)

[5 Requirements Gathering & Analysis 20](#_Toc511578205)

[Requirements Gathering 20](#_Toc511578206)

[Analysis 20](#_Toc511578207)

[6 Design 20](#_Toc511578208)

[7 Implementation 20](#_Toc511578209)

[System Architecture 20](#_Toc511578210)

[Detail code 20](#_Toc511578211)

[8 Challenges and Solutions 21](#_Toc511578212)

[9 Testing and Results 21](#_Toc511578213)

[10 Evaluation 21](#_Toc511578214)

[11 Conclusions 21](#_Toc511578215)

[12 Suggested Future Improvements to Project 21](#_Toc511578216)

[References 22](#_Toc511578217)

[Appendices 24](#_Toc511578218)

Table of Figures

[Figure 1 - Social Networking Tracker (Jan 2017) 3](#_Toc511578219)

[Figure 2 - Social Messaging Tracker (Jan 2017) 4](#_Toc511578220)

[Figure 3 - Roof.ai chatbot example (Moubarak, 2018) 9](#_Toc511578221)

[Figure 4 - Roof.ai widget on The Keyes Company website homepage (The Keyes Company, 2018) 9](#_Toc511578222)

[Figure 5 - The Keyes Company example conversation with Roof.ai chatbot 10](#_Toc511578223)

[Figure 6 - Lemonade chat example (Lemonade, 2018) 11](#_Toc511578224)

[Figure 7 - Example of Lemonade question with options (Lemonade, 2018) 12](#_Toc511578225)

[Figure 8 - Software Development Life Cycle (Gordiyenko, 2014) 17](#_Toc511578226)

[Figure 9 - Diagram of Spiral Model (Boehm, 1988) 18](#_Toc511578227)

[Figure 10 - Representation of the Agile Development Model (ISTQB Exam Certification, 2018) 19](#_Toc511578228)

# 1 Introduction

## 1.1 Background

Applied Systems have operated for over 30 years powering the insurance industry across the USA, Canada, Ireland and the United Kingdom, providing industry leading technology to the insurance industry (Applied Systems, 2018).

As a company; Applied Systems strive to provide insurance brokers with innovative software solutions to maximise the brokers business profits and improve customer communications.

As an employee of Applied Systems, and being sponsored by then during my university studies, the challenge was set for me to use my Computing Systems Project as an opportunity to research and implement a proof-of-concept for the next innovative piece of software that they may put in to production for release to the market.

Considering my background as a software developer with experience working on web-based products, I decided to research emerging trends on the web in relation to business-to-customer interactions in the Insurtech (insurance technology) industry.

My research found that the number one predicted trend in the Insurtech market as predicted by “The Digital Insurer” is “automation will replace human effort across the entire insurance value chain” (Huckstep, 2017). Although this is not a trend restricted to the insurance industry it is likely to have quite a significant impact in the insurance world as a lot of the industry business methods continue to operate in a manner that is outdated and is more suited to consumers prior to the Internet. With the rise of the Internet, consumers now want a full digital experience without need for human interaction (Huckstep, 2017).

This theory of consumers becoming more “digital” is backed up by looking at the PWC Irish Total Retail Survey where they found that 48% of Irish consumers had used their mobile phones to shop online at least a few times a year, with 30% stating that they feel their mobile will become their main method of shopping in the future (PwC, 2017).

Jay Samit at Fortune also predicts 2018 to be “the year of the bots” (Samit, 2017). Bots will become more intelligent in the use and understanding of natural language to become more capable of helping us with our daily routines.

As well as the predicted growth of chatbots, Ipsos MRBI Tracker research shows 64% of people in Ireland are using Facebook as a social media platform (Figure 1), with 58% using Facebook Messenger as their social media messenger app of choice (Figure 2). This gives me confidence that building my chatbot for release to Facebook Messenger is the correct strategy.

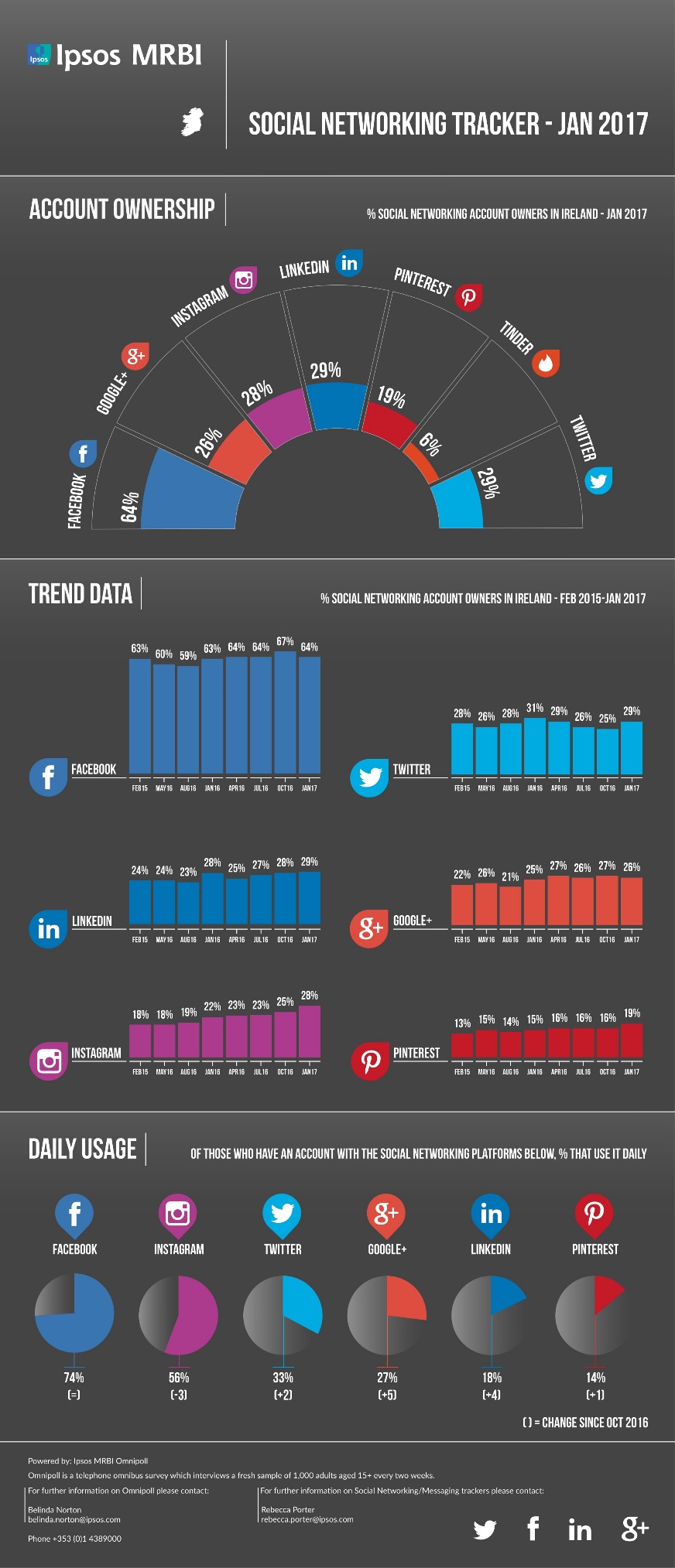


Figure 1 - Social Networking Tracker (Jan 2017)

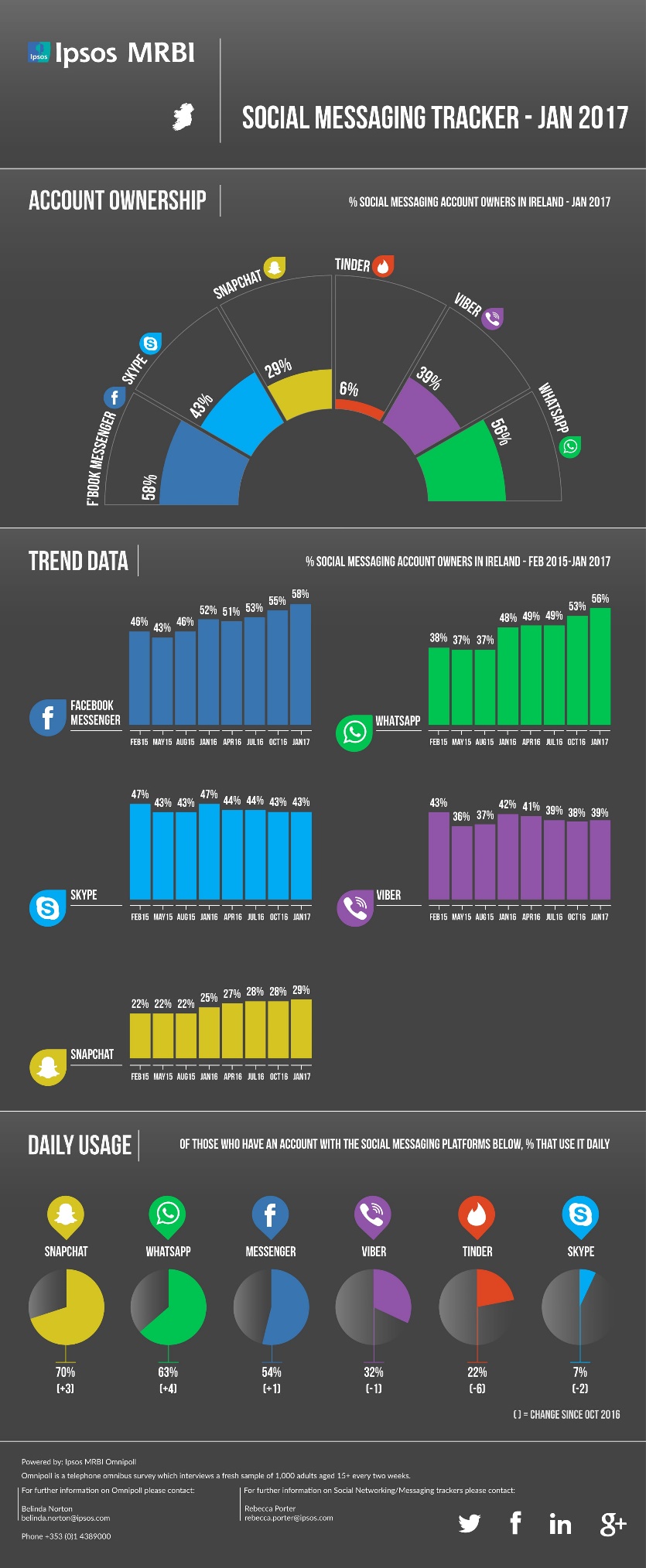


Figure 2 - Social Messaging Tracker (Jan 2017)

With the findings from the research and taking my personal interest in to account, I have decided to develop a chatbot that can be deployed to use on Facebook Messenger. The chatbot should enable insurance broker customers to communicate with the bot and receive an insurance quote.

## 1.2 Project Aims

The aim of the project is to create a proof-of-concept chatbot for Applied Systems.

The chatbot should allow Facebook Messenger users to “chat” with the bot and receive a motor or home insurance quote based on the information they have entered.

## 1.3 Project Objectives

To give the project structure, a list of project objectives has been identified and listed below:

* Allow the user to get a motor insurance quick quote[[1]](#footnote-1)
* Allow the user to get a home insurance quick quote1
* Keep a record of conversations between the user and the bot
* Keep a record of any quotes returned to the user
* Keep a record of any errors encountered during the workflow
* Allow user to retrieve a previous quote
* Allow user to choose a returned quote
* Allow user to choose to be contacted by the insurance broker that provided the quotes
* Follow up conversation with email to insurance broker and the user

## 1.4 Project Activities

To ensure that as many objectives as possible are met, a list of project activities has been identified and listed below:

* Determine questions and question order for motor insurance quick quote
* Determine questions and question order for home insurance quick quote
* Design database for storing conversations, quotes and errors
* Establish possible SQL queries for inserting, updating and reading from database
* Research possible framework to use for implementing chatbot
* Research best programming language to use
* Create chatbot solution
* Create logging service
* Create database
* Write and implement database
* Write and implement test plan
* Perform user testing
* Implement changes from user testing

## 1.5 Outline of Dissertation Structure

The rest of this dissertation will follow the below structure:

* Chapter 1 – Introduction

Give an overview of how the project came to life. The project aim will be outlined with the project objectives and aims listed.

* Chapter 2 – Chatbots in Use Today

An exercise evaluating chatbots currently in operation. A variety of chatbots will be identified and critiqued; highlighting the advantages of using a chatbot can have for a business.

* Chapter 3 – Technical Background

A review of possible technology that could be used to build the chatbot with a reasoning given for using the tech stack that will be used.

* Chapter 4 – Development Lifecycle

The Software Development Lifecycle to be used for this project will be explained.

* Chapter 5 – Requirements Gathering and Analysis

Explanation of how requirements for the project where determined. The project requirements will be detailed alongside a risk analysis of the project.

* Chapter 6 – Design

An overview of the system design at a high level, the initial database design and the design of the chatbot conversations between the user and the bot.

* Chapter 7 – Implementation

A detailed assessment of the system implementation. Programming practices, libraries and frameworks used, and code snippets will be detailed and explained in this chapter.

* Chapter 8 – Challenges and Solutions

A review of some of the challenges faced during the project lifecycle with solutions to these challenges noted.

* Chapter 9 – Testing and Results

A chapter on the testing techniques used during the project with results of the tests given.

* Chapter 10 – Evaluation

An evaluation of the project management techniques used, system implementation, technology used, and personal experience gained throughout the project.

* Chapter 11 – Conclusions

A lookback on the project and its successes and failures.

* Chapter 12 – Suggested Future Improvements to Project

A look at how the project could be improved in the future with reference to additional functionality, implementation and project management processes.

# 2 Chatbots in Use Today

## 2.1 What is a chatbot?

Business Insider UK defines a chatbot as a robot that can maintain a conversation with a human. It is essentially a virtual conversation with a piece of software (Nguyen, 2017).

## 2.2 Review of Chatbots

To get a better understanding of how a chatbot should be designed and how users typically interact with them, a critical evaluation of a selection of chatbots from various business sectors has been performed with the findings detailed below.

The chatbots I have selected for review are:

* Marvel
* RoofAi
* Lemonade

### 2.2.1 Marvel

The Marvel chatbot is a way for fans of Marvel Comics to chat directly to some of their favourite characters through Facebook Messenger or Twitter DM (Morse, 2017).

The chatbot was built with Conversable, a platform for building AI-enhanced messaging experiences (Conversable, 2018), making use of natural language processing (NLP) and machine learning which bots are more frequently using.

The Marvel bot unfortunately is not a true conversationalist. The conversation has a couple of paths that the conversation can go down but always ending in the same way – trying to sell the user merchandise. Which admittedly is a great business use but could be frustrating for the end user if they are inevitably going to end every conversation with an advertisement enticing them to spend money.

### 2.2.2 RoofAi

RoofAi promotes itself as a “smart chat” bot – “a combination of live chat and bots” (RoofAi, 2018). RoofAi view their website widget, a widget built for realtors in the United States, as a tool to manage real-time conversations as and when your website user has a query – stating that “you can now be instantly responsive 24/7, 365” (Moubarak, 2018).

The Roof.ai bot is capable of capturing user information, scheduling viewing appointments and directing leads to the best suited real estate agent; e.g. assigning a user looking to rent a property to an available agent responsible for rental properties (Moubarak, 2018).

The bot is implemented on a customer site in the form of widget.

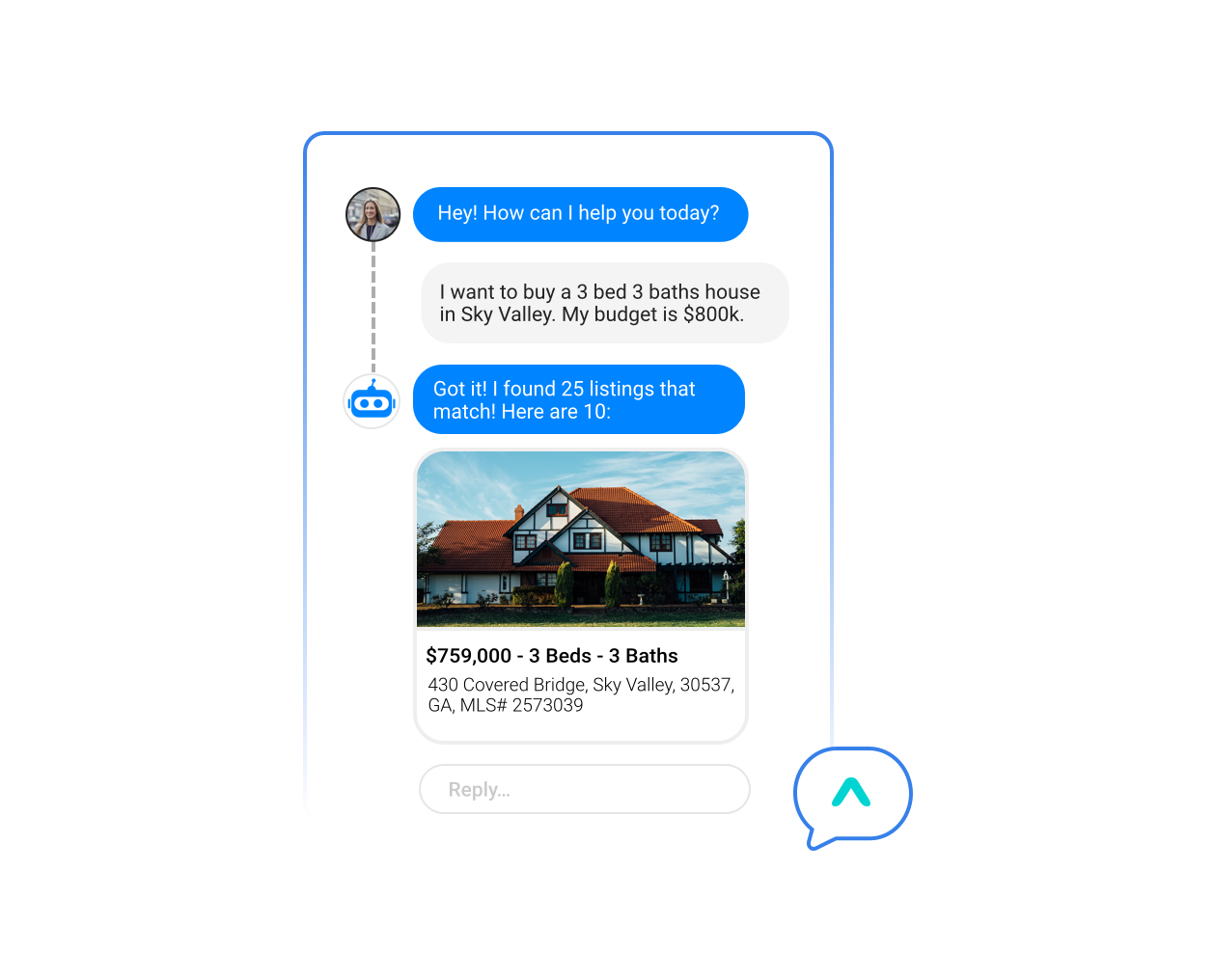


Figure 3 - Roof.ai chatbot example (Moubarak, 2018)

#### 2.2.2.1 Roof.ai Review

Using the website for The Keyes Company, a Florida based real-estate agency, they have added the Roof.ai chatbot widget to their homepage. The widget has been styled to fit with the website and makes use of The Keyes Company logo and branding colours.

It is immediately offering assistance as can be seen below. This is good as it draws attention to the bot in a subtle manner and highlights that the bot is actively available.

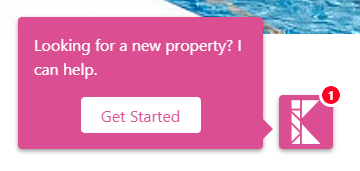


Figure 4 - Roof.ai widget on The Keyes Company website homepage (The Keyes Company, 2018)

Having used the bot to search for properties in the Florida area the language the bot uses is a positive standout. The language is professional but friendly.

The conversation is guided by the bot asking questions as a human estate agent would. With some questions the bot gives the user some choices which adds an extra level of guidance and validation to ensure that answers given are correct and match with expected answers.

Questions are asked immediately after an answer has been received so the communication is fast and direct.

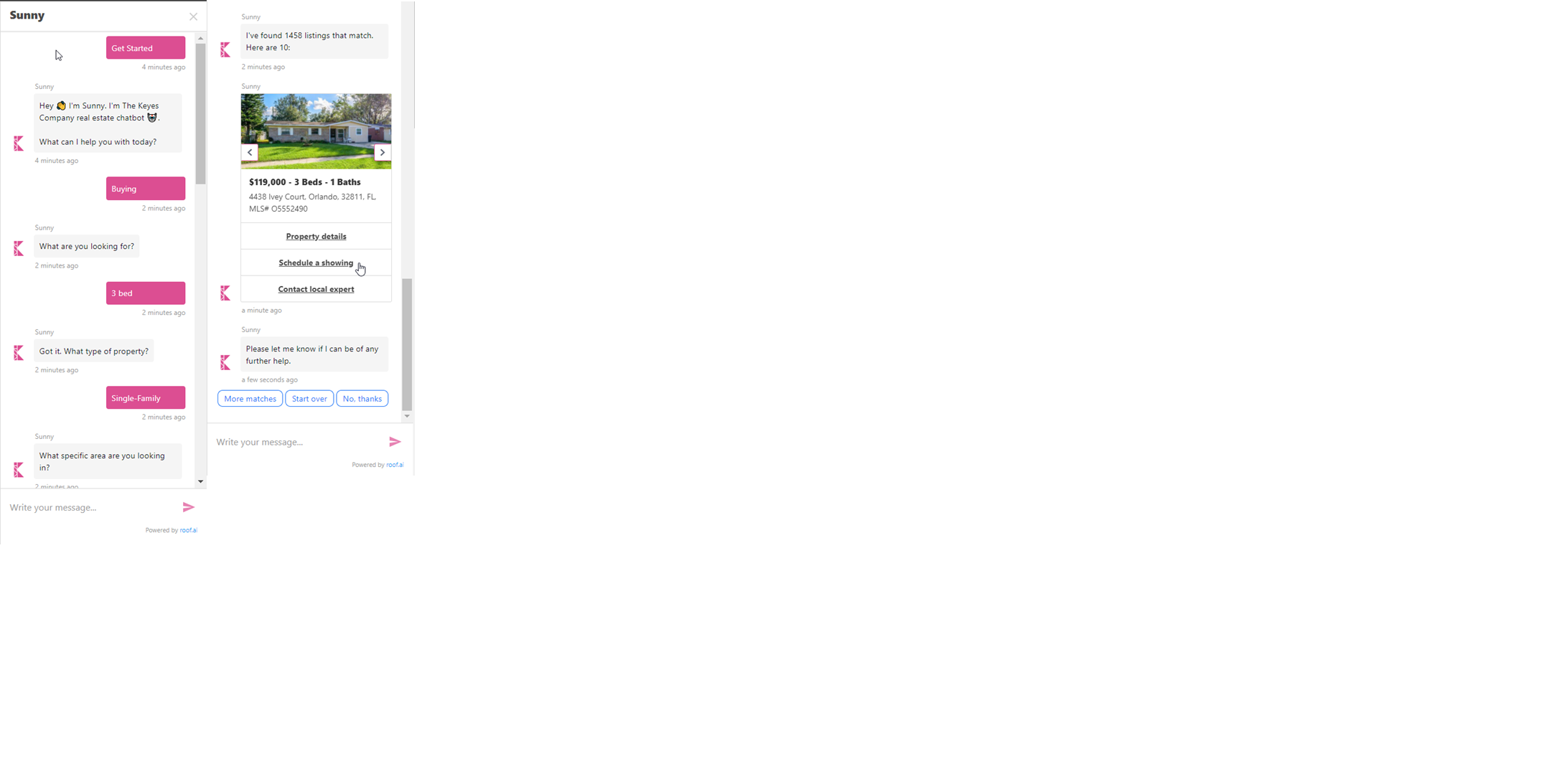


Figure 5 - The Keyes Company example conversation with Roof.ai chatbot

As can be seen above the chatbot was able to return a selection of properties based on the criteria entered. The properties are nicely displayed in a scrollable carousel making it easy for the user to browse the properties.

Similarities between the Roof.ai chatbot and the chatbot being developed in this project are the use of guided conversation. The benefit of this being that the user should not go too far off the main purpose of the chatbot. The Roof.ai chatbot also does a good job of giving the user options for answering questions but mixing that with questions open to a free response so the chat feels more natural, this is also something to be implemented in this project’s chatbot.

### 2.2.3 Lemonade

The chatbot that also currently operates in the insurance industry is used, and was developed by, a company called Lemonade Insurance Agency based in New York.

Lemonade tell their users to “Forget everything you know about insurance” (Lemonade, 2018), as they sell insurance based on a new business model and a central component to this is their artificial intelligence bot (Wissner-Levy, 2016). They aim to make the process of getting insurance faster, more honest and more transparent (Wissner-Levy, 2016).

Lemonade also offer a widget and API for integration on other websites or applications (Lemonade, 2018).

#### 2.2.3.1 Lemonade Review

Through the Lemonade website ([www.lemonade.com](http://www.lemonade.com)), a review of the process to get an insurance price was completed.

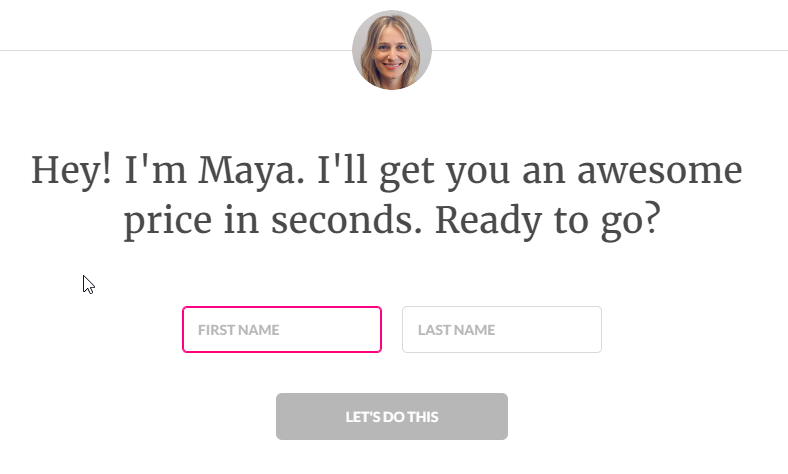


Figure 6 - Lemonade chat example (Lemonade, 2018)

Lemonade have given their bot a name and a profile picture, Maya (Lemonade, 2018). This adds a personal touch to the bot and makes it feel less like chatting to a piece of software and more like chatting to a human being which will have a positive impact on the end user.

Similarly to Roof.ai, the Lemonade chatbot asks questions in a manner that guides the user through the workflow whilst providing options to answer the questions, as below.

The language used by the bot again is natural and polite.

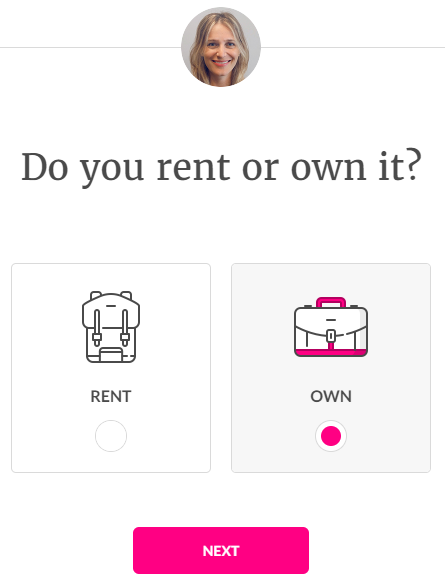


Figure 7 - Example of Lemonade question with options (Lemonade, 2018)

# 3 Technical Background

Before development; research has been carried out on how to best build a chatbot and complete the project to meet all the project objectives. The review will include; development languages, any frameworks or libraries that could be utilised, database options and development environments.

Options for source control will also be discussed.

## 3.1 Frameworks

As with development of any project, it is possible to use frameworks to aid development. Using frameworks can quicken development. Wikipedia defines a framework as being “a universal, reusable software environment that provides particular functionality as part of a larger software platform to facilitate development of software applications, products and solutions” (Wikipedia, 2018).

A framework, in the context of a chatbot, i.e. a bot framework, is a great tool to have as it abstracts a lot of the manual work involved in creating a bot (Maruti Techlabs, 2018).

To develop a chatbot, the following bot frameworks have been considered:

* Microsoft Bot Framework
* Wit.ai
* DialogFlow

### 3.1.1 Microsoft Bot Framework

The Microsoft Bot Framework is comprised of various tools in its software development kit (SDK). The main tools are:

* Bot Connector
* LUIS

#### 3.1.1.1 Bot Connector

This service in the Microsoft Bot Framework is what enables the bot to communicate messages on channels (the platform the bot is configured to run on, e.g. Facebook Messenger, Slack, Skype, etc.) (Microsoft, 2017).

Communication on these channels is achieved using industry-standard REST and JSON over HTTPS.

#### 3.1.1.2 LUIS.ai

Meaning Language Understanding Intelligent Service – LUIS enables a bot to understand natural language when input by the user (Berry, 2017). It uses machine learning to accept input and extract an intended meaning from the input so it can return a relevant response (Berry, 2017).

The key concepts to LUIS are:

* Intents – these can be considered the “action” a user wishes to perform within an application. Within LUIS a language model, the developer defines intents and maps these to actions (Berry, 2017)
* Utterances – these are the text the user may input that the bot needs to be able to receive and understand (Berry, 2017). There can be many variations to an utterance but the utterance will be linked to a specific intent.
* Entities – these are pieces of information that may appear in an utterance. Identifying entities in an utterance, LUIS is able to choose the best suited action to response to the user (Berry, 2017).

In using LUIS, a developer defines a domain specific language model and fills it with intents, utterances and entities (Berry, 2017). The model then must be trained and published. The LUIS app will then receive an utterance as a HTTP request. From this request it determines the user interaction and responds (Berry, 2017). The user utterance sent from the client application is evaluated to a JSON object by LUIS which is then sent back to the client app (Berry, 2017).

### 3.1.2 Wit.ai

Wit.ai is an open-source API (Application Programming Interface) that “makes it easy for developers to build applications and devices that you can talk or text to” (wit.ai, 2018). It is a natural language platform that uses each interaction to learn so it can provide more accurate responses (wit.ai, 2018).

One appealing aspect of Qit.ai is that because it is open-source, it is able to share what it has learned across the all developers using wit.ai.

Wit also uses entities and intents to understand the action the user is trying to perform.

### 3.1.3 DialogFlow

On the same premise as Microsoft’s LUIS and Wit.ai; DialogFlow uses machine learning to understand meaning from what a user has input or said.

DialogFlow use an “agent” to manage the conversation between the user (human) and bot (DialogFlow, 2018). DialogFlow describe these agents as Natural Language Understanding (NLU) modules (DialogFlow, 2018). The NLU module converts the user input into data that can determine an action.

Other important aspects of DialogFlow are:

* Entities – domain-specific phrases that can be mapped to NLP (Natural Language Processing) phrases (Maruti Techlabs, 2018)
* Intents – the action to be taken based on what a user has input (Maruti Techlabs, 2018)
* Actions – what will happen based of the identified intent (Maruti Techlabs, 2018)
* Contexts – a string representation to evaluate the user expression. Useful for determining meaning (Maruti Techlabs, 2018)

## 3.1.4 Messaging Platforms Available Using Frameworks

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Framework allows for communication on platform? | | | | | | |
|  | Facebook Messenger | Skype | Slack | Microsoft Teams | Telegram | Kik | Cortana |
| Microsoft Bot Framework |  |  |  |  |  |  |  |
| Wit.ai |  |  |  |  |  |  |  |
| DialogFlow |  |  |  |  |  |  |  |

# 4 Development Lifecycle

The Software Development Life Cycle is a process that aims to produce software with the highest quality and with the lowest cost possible in the shortest amount of time (Stackify, 2017).

According to (Stackify, 2017), the advantages of following the SDLC are:

* It allows a high level of management control
* Gives developers a good understanding of what they are trying to build
* An agreement is made upfront on what the project outcome should be
* It sets out an agreed plan on how to reach the proposed goal

The Software Development Life Cycle defines six stages that can be implemented in various ways by different SDLC models.

These 6 stages are:

* Planning
* Defining
* Designing
* Building
* Testing
* Deployment

The models, also called Software Development Process Models, will follow steps unique to each model but will still all either strictly or loosely follow the Software Development Life Cycle stages (Tutorials Point, 2017).

## 4.1 Software Development Process Models Considered

The following models have been considered as a development model to use on this project.

### 4.1.1 Waterfall Model

The first process model to be introduced to the software development industry, it was designed to be used in a wat that processes do not overlap; one process must finish before the other starts (Tutorials Point, 2017).



Figure 8 - Software Development Life Cycle (Gordiyenko, 2014)

#### 4.1.1.1 Advantages of Waterfall Model

* Simple to use and understand
* Each process has specific goals and outcomes
* Stages of the project are well defined

#### 4.1.1.2 Disadvantages of Waterfall Model

* A working product is not delivered until near the end of the life cycle
* Not suitable for projects with requirements that are likely to change
* Stages must wait on their predecessors to finish before they can start

#### 4.1.1.3 Why the Waterfall model was not chosen

Due to its strict phase completion rules, the Waterfall model is not suited for this project that is open to changing requirements throughout the duration of the project.

The plan is also to have regular feedback from stakeholders and users which again does not fit with the Waterfall model process.

### 4.1.2 Spiral Model

Consisting of four phases; planning, Risk, Engineering and Evaluation, a software project using this model will pass through each phase iteratively until the project is delivered (International Software Testing Qualifications Board, 2017).

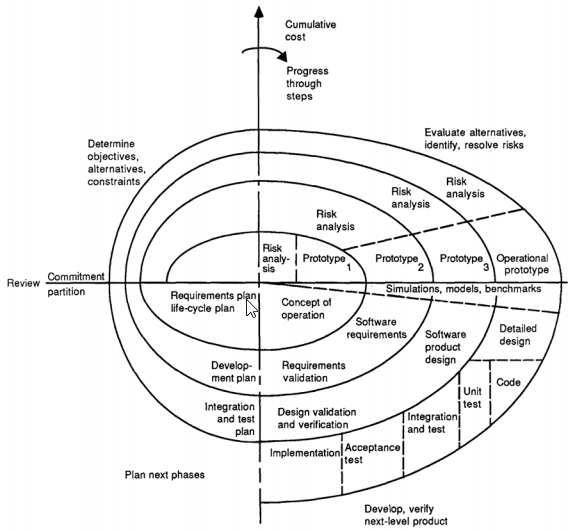


Figure 9 - Diagram of Spiral Model (Boehm, 1988)

#### 4.1.2.1 Advantages of Spiral Model

* Emphasis on risk analysis means risk are identified early and can be managed or avoided
* Software is produced early and frequently
* Software functionality can change or be added late in to the project

#### 4.1.2.2 Disadvantages of Spiral Model

* Can be an expensive model
* Not suited to small projects
* Risk analysis requires experienced analysts

#### 4.1.2.3 Why the Spiral model was not chosen

The Spiral model was not selected for use on this project due to the short life span of the project. Spiral is more suited to larger projects with a long-term commitment.

The project is also considered low risk and the requirements of the project are clear and concise.

### 4.1.3 Agile Model

The Agile SDLC model put focus on delivering products to the customer on a regular basis. It combines iterative and incremental models and focuses on customer satisfaction and welcomes change (Tutorials Point, 2017).

A project using the agile model will break a project down in to smaller pieces of work and deliver these in an iterative manner.

Each iteration is able to have multiple processes active at any one time. The processes include:

* Planning
* Requirements Analysis
* Design
* Coding
* Unit testing
* Acceptance testing

The iterations that produce working software are usually time boxed to an amount of time decided by the team.

Iterations can happen in a similar way to the image below.



Figure 10 - Representation of the Agile Development Model (ISTQB Exam Certification, 2018)

The Agile model come with an Agile Manifesto (Agile Manifesto, 2017) that states the following principles:

* Individuals and interaction over processes and tools
* Working software over comprehensive documentation
* Customer collaboration over contract negotiations
* Responding to change over following a plan

#### 4.1.3.1 Advantages of Agile

* A realistic take on how software is best developed
* Functionality is developed quickly
* Processes work with pre-defined or changing requirements
* Little or no planning required

#### 4.1.3.2 Disadvantages of Agile

* Depends heavily on stakeholder and customer communication and feedback
* Less focus on documentation can lead to problems when on boarding new team members or handing a project on to another team
* Changing requirements and functionality can have an adverse effect on the project delivery deadline and can lead to scope creep on the project

#### 4.1.3.3 Why Agile was chosen for this project

Agile has been chosen as the software development model for the project as the Agile model is one widely practised within Applied Systems.

The opportunity to develop software quickly and get regular feedback on it means the end product is more likely to meet the Project Sponsor’s expectations.

Not having to focus on detailed documentation also means the emphasis can be placed on building a working system. Due to the tight schedule of this project, this is a benefit that cannot be overlooked.

# 5 Requirements Gathering & Analysis

## Requirements Gathering

## Analysis

# 6 Design

# 7 Implementation

## System Architecture

## Detail code

*Code Explanation*

*Libraries used*

# 8 Challenges and Solutions

# 9 Testing and Results

*Validation and verification*

*Evaluation of process and methods used to reach outcome*

# 10 Evaluation

Fulfilment of project objectives

# 11 Conclusions

# 12 Suggested Future Improvements to Project

# References

Agile Manifesto. (2017, 10 29). *Manifesto for Agile Software Development*. Retrieved from Agile Manifesto website: www.agilemanifesto.org

Applied Systems. (2018, 04 11). *About Us - About Applied*. Retrieved from Applied Systems website: https://www1.appliedsystems.com/en-ie/about-us/about-applied/

Berry, D. (2017, March 03). *About Language Understanding (LUIS)*. Retrieved April 15, 2018, from Microsoft Azure: https://docs.microsoft.com/en-us/azure/cognitive-services/LUIS/Home

Boehm, B. W. (1988). *A Spiral Model of Software Development and.* TRW Defense Systems Group. Retrieved April 14, 2018, from http://csse.usc.edu/TECHRPTS/1988/usccse88-500/usccse88-500.pdf

Conversable. (2018). *About Conversable*. Retrieved April 14, 2018, from Conversable: http://conversable.com/about/

DialogFlow. (2018, February 21). *Agents*. Retrieved April 15, 2018, from DialogFlow: https://dialogflow.com/docs/agents

Gordiyenko, S. (2014, November 3). *Software Development Life Cycle (SDLC)*. Retrieved April 14, 2018, from XB Software: https://xbsoftware.com/blog/software-development-life-cycle-waterfall-model/

Huckstep, R. (2017, December 13). *10 Insurtech Trends at the Insurance Crossroads 2018*. Retrieved April 13, 2018, from The Digital Insurer: https://www.the-digital-insurer.com/blog/10-insurtech-trends-insurance-2018/

International Software Testing Qualifications Board. (2017, 10 29). *What is Spiral model- advantages, disadvantages and when to use it?* Retrieved from ISTQB website: http://istqbexamcertification.com/what-is-spiral-model-advantages-disadvantages-and-when-to-use-it/

ISTQB Exam Certification. (2018). *What is Agile model – advantages, disadvantages and when to use it?* Retrieved April 14, 2018, from ISTQB Exam Certification: http://istqbexamcertification.com/what-is-agile-model-advantages-disadvantages-and-when-to-use-it/

Lemonade. (2018). *Lemonade Renters and Home Insurance*. Retrieved April 15, 2018, from Lemonade: https://www.lemonade.com/

Lemonade. (2018). *Sign up for Lemonade Renters & Home Insurance*. Retrieved April 15, 2018, from Lemonade: https://www.lemonade.com/start/1

Lemonade. (2018). *The Lemonade Insurance API*. Retrieved April 15, 2018, from Lemonade: https://www.lemonade.com/api

Maruti Techlabs. (2018). *Complete Guide on Bot Frameworks*. Retrieved April 15, 2018, from Maruti Techlabs: https://www.marutitech.com/complete-guide-bot-frameworks/

MedWhat. (2017). *About Us*. Retrieved April 14, 2018, from MedWhat: https://medwhat.com/about-us/index.html

Microsoft. (2017). *Mircosoft Bot Framework*. Retrieved April 15, 2018, from Bot Framework: https://dev.botframework.com/

Morse, B. (2017, April 20). *Become Part of the Story with the Marvel Comics Chatbot*. Retrieved April 14, 2018, from Marvel: https://news.marvel.com/comics/63529/become-part-story-marvel-comics-chatbot/

Moubarak, M. (2018, February 16). *Smart Chat For Your Real Estate Website: Qualify Buyers And Sellers in Real-Time With Roof.ai*. Retrieved April 15, 2018, from roof.ai: https://blog.roof.ai/roofai-real-estate-chatbot/

Nguyen, M.-H. (2017, October 6). *How artificial intelligence & machine learning produced robots we can talk to*. Retrieved April 14, 2018, from Business Insider UK: http://uk.businessinsider.com/what-is-chatbot-talking-ai-robot-chat-simulators-2017-10

PwC. (2017). *PwC Irish Total Retail 2017 Survey*. Retrieved April 13, 2018, from PwC Irish Total Retail 2017 Survey: https://www.pwc.ie/publications/2017/ten-retailer-investments-for-an-uncertain-future-total-retail-survey.pdf

RoofAi. (2018). *Smart chat for your real estate website*. Retrieved April 14, 2018, from RoofAi: https://www.roof.ai/

Samit, J. (2017, December 26). *4 Technology Trends That Will Transform Our World in 2018*. Retrieved April 13, 2018, from Fortune: http://fortune.com/2017/12/26/4-technology-trends-2018/

Stackify. (2017, 10 04). *What is SDLC? Understand the Software Development Life Cycle*. Retrieved from Stackify website: https://stackify.com/what-is-sdlc/

The Keyes Company. (2018). *Florida Real Estate & Homes For Sale*. Retrieved April 15, 2018, from The Keyes Company: https://www.keyes.com/

Tutorials Point. (2017, 10 29). *SDLC Agile Model*. Retrieved from Tutorials Point website: https://www.tutorialspoint.com/sdlc/sdlc\_agile\_model.htm

Tutorials Point. (2017, 10 29). *SDLC Overview*. Retrieved from TutorialsPoint: https://www.tutorialspoint.com/sdlc/sdlc\_overview.htm

Wikipedia. (2018, April 3). *Software framework*. Retrieved April 15, 2018, from Wikipedia: https://en.wikipedia.org/wiki/Software\_framework

Wissner-Levy, Y. (2016, September). *How Lemonade is Fixing a Broken System*. Retrieved April 15, 2018, from Lemonade: https://www.lemonade.com/blog/hello-world/

wit.ai. (2018). *Wit.ai*. Retrieved April 15, 2018, from wit.ai: https://wit.ai/

# Appendices

1. A ‘quick quote’ is a reduced set of questions that still allows insurance quotes to be returned when requested. For questions not asked, default answers are set. [↑](#footnote-ref-1)