

Interim Report Domain Specific Language for Social Media Bots

Samuel Withey
Supervisor: Ian Wakeman

Contents

1	Interim-Report	4
1.1	Introduction	4
1.1.1	Goals, Aims and Objectives	4
1.1.2	Problem Area	4
1.1.3	Motivation	8
1.1.4	Relevance	9
1.2	Professional Considerations	10
1.2.1	US Presidential Election 2016	10
1.2.2	BCS Code of Conduct	10
1.3	Related Work	12
1.3.1	TweetDeck	12
1.3.2	Hootsuite	13
1.3.3	Social Bot	15
1.3.4	Software Bots	15
1.3.5	Research	16
1.4	Requirements Analysis	17
1.4.1	Functional Requirements	17
1.4.2	Non-Functional Requirements	19
1.4.3	Requirements	20
1.4.4	High Level Diagrams	20
1.5	Project Plan	23
1.5.1	Task 1: Proposal and Interim Report	23
1.5.2	Task 2: Domain Specific Language	23
1.5.3	Task 3: Interpreter	24
1.5.4	Task 4: Web Application	25
1.5.5	Task 5: Final Report	26
1.5.6	Gantt Chart	26
2	Appendix	29
2.1	Log	29
2.1.1	Meeting 1: 03/10/2019	29
2.1.2	Meeting 2: 16/10/2019	29
2.1.3	Meeting 3: 24/10/2019	30
2.1.4	Meeting 4: 11/11/2019	31

2.2	Proposal	33
2.2.1	Goals	33
2.2.2	Objectives	33
2.2.3	Relevance	35
2.2.4	Resources Required	35
2.3	Ethical Consent	36

1 Interim-Report

1.1 Introduction

In January 2019, according to Hootsuite's digital report, there were 3.484 billion world-wide active users across all social media platforms [2]. This rapid social paradigm shift has transformed consumer behaviour and social interaction through online social platforms which allow instantaneous direct communication. This has provided unbounded limitations for brands to connect and engage directly with their audiences.

1.1.1 Goals, Aims and Objectives

The overall goal of this project is to provide a solution to the issues around managing social media content. This solution will be achieved by providing tools to automate managing and posting content on social media platforms. The aims for these tools are to allow the user to post content for a single account on multiple social media platforms at once, schedule these posts and automate replies and direct messages to other users on the platforms.

The solution for the project can be split into three main objectives. The first objective is to design a system or tool to provide the user with methods to automate the features and capabilities of what can be done by a user on social media platforms. The second objective is to design a system or tool to execute these features and capabilities of the first objective and the third objective is to design a centralised system to display customisable feeds for multiple social media platforms.

1.1.2 Problem Area

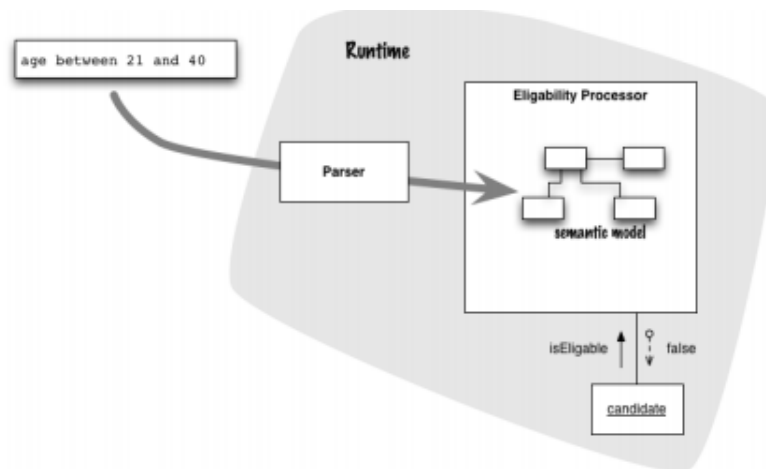
The first objective will be achieved by designing and implementing a domain specific language and parser to be able to configure and automate the features on social media platforms. The second objective will be achieved by implementing an interpreter to translate the domain specific language into executable pre-compiled code to interact with the social media platforms. The third objective will be achieved by implementing a web application which displays customisable feeds for multiple social media accounts

in a centralised interface.

Domain specific languages are optimised for a given class of problems called a domain. It is based on abstractions that are closely aligned with the domain for which the language is built and a domain specific languages syntax is suitable for expressing these abstractions concisely. This differs from general-purpose languages which are used by programmers to instruct computers. General-purpose languages are Turing complete, which means they can be used to implement anything that is computable by a Turing machine. [16]

An advantage of using a domain specific language is that it is another layer of abstraction. Notation can be defined that expresses the abstractions concisely and makes interacting with programs easy and efficient. This is important for the project as the domain specific language can be used by non-programmers. It provides a clean level of abstraction that moves away from general-purpose languages and API's which non-programmers are not competent enough to use and allows them to work with a language closely aligned with the domain they work in [16]. This is a key solution to the problem area as it abstracts the complexity of using API's by creating a language which is syntactically simpler than a general purpose language and can be used by non-programmers which is the target user for the system.

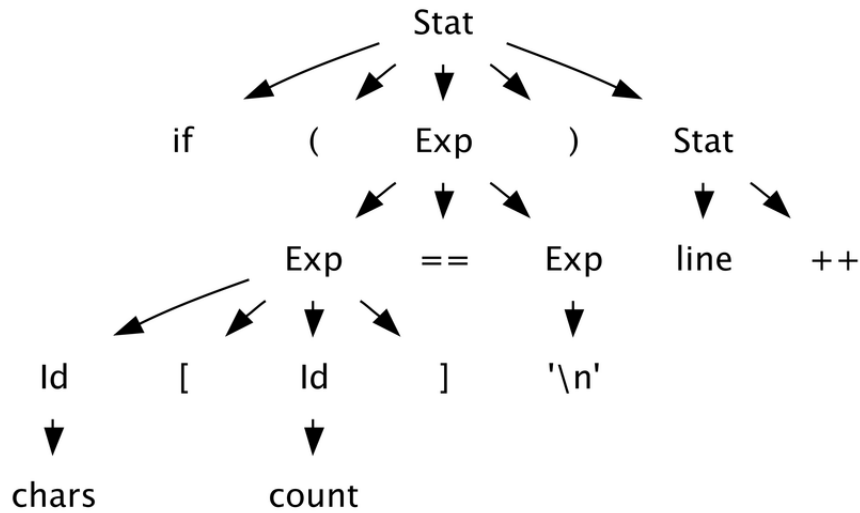
For the domain specific language to execute it requires an interpreter or compiler. When text is interpreted, it is parsed and the result from the program is produced in a single process.



Interpreter parsing text and produces result [9]

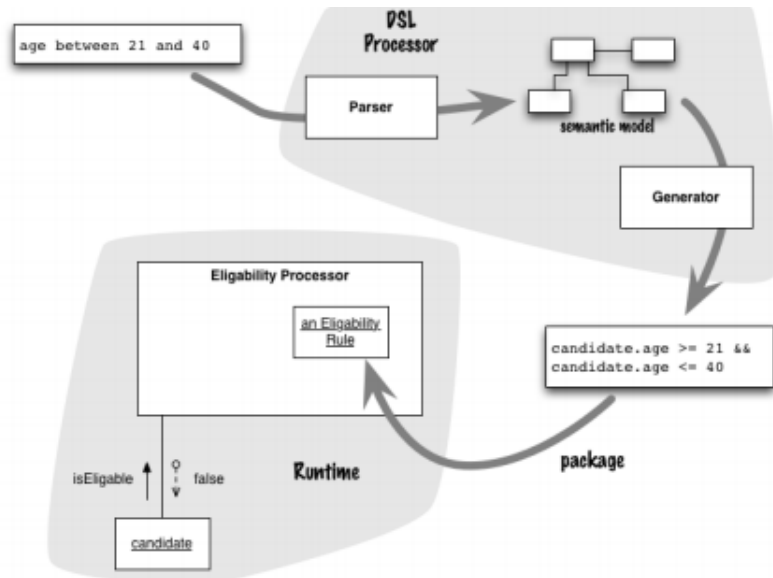
A parser is the processing of structuring a text according to a given grammar. The parser will generate a syntax/parse tree which is a data structure which precisely shows how various segments of the program text are to be viewed in terms of the grammar. A grammar or context-free grammar are the formalism for describing the structure of

programs in a programming language by describing the syntactic structure. Since the semantics of a language is defined in terms of the syntax, the context-free-grammar is also instrumental in the definition of the semantics [12].



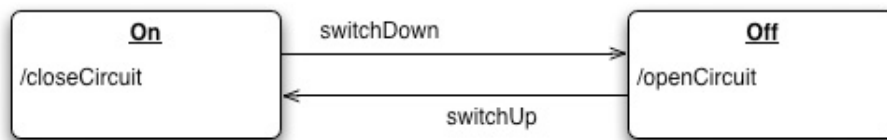
Java Parse Tree [8]

The alternative to interpretation is compilation. Compilation parses a program text and produces an intermediate output, which is then separately processed to provide desired behaviour. For domain specific languages the compilation is often referred to as code generation. An advantage of code generation is that code can be generated in any programming language. This is useful for the project as the code generated would interact with the API for the social media platforms [9].



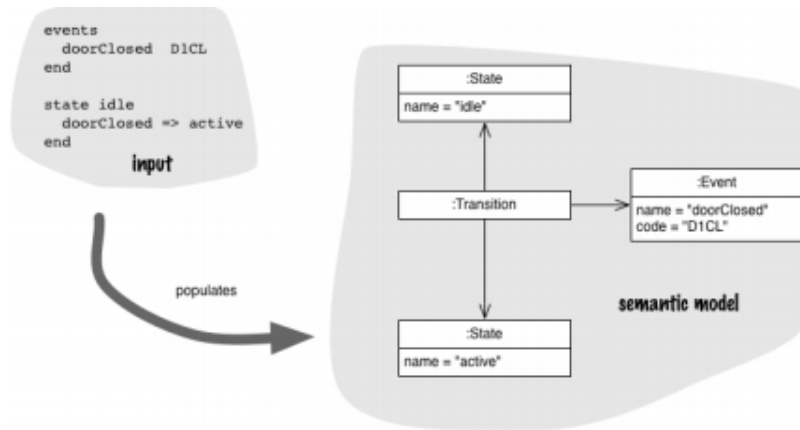
Compiler parsing text [9]

A state machine is a model of a system as a set of explicit states with transitions between them. Internal states can be classified and the system can move between states depending on some internal property and event; thus a sequence of events leads from state to state [9].



State Machine [9]

A semantic model is a state machine which captures the legal expressions of a program and the model defines the semantics. For a semantic model to work, the domain specific language provides a readable way of populating the model and acts as a mechanism for how the model is configured [9].



Parsing DSL populates semantic model [9]

Semantic models can be implemented into the compilation stage which allows the separation of the parsing, execution semantics and the code generation. An advantage of using semantic models is the interpreter and code generation can both be completed from the same semantic model which makes the process simpler. The domain specific language acts as a front-end to the semantic model which provides a different style of manipulation to the command-query API [9].

1.1.3 Motivation

The motivation for this project was to provide a unique solution for social media automation and management. The project scope covers multiple compelling areas within computing such as compilers and interpreters, software engineering, web applications and distributed systems. Other motivations for the project includes the opportunity to use industry standard tools and applications such as parser generators, application programming interfaces, version control systems, web frameworks and cloud computing services.

The project was inspired by previous industry experience working for a large tech company which provided tools to solve issues around paid advertisement. This inspiration led to a desire to explore unique solutions to manage and automate organic social media content and interactions for multiple social media platforms simultaneously. Other industry experience working on intricate web applications inspired the projects social media management tool to be created using Python's Web Framework Django.

The ultimate motivation for the project is to apply and continue to develop the skills learnt in previous job roles and at higher education into compelling areas of computing for future graduate job roles.

1.1.4 Relevance

The project relevance extends to various areas in computing that have been studied at higher education and have been applied in prior work experience. The main area in computing that this project explores is the domain specific language and interpreter. This area of computing has been previously taught through Compilers and Computer Architecture module at University. This is an important research topic within Computer Science and will be useful to apply the knowledge from this subject area to a practical application.

Another area of computing that this project explores is Software Engineering. This subject area is relevant to the project as the domain specific language will require rigorous design to ensure that it is scalable, efficient and usable and will require more traditional waterfall planning and techniques. The development of the domain specific language, interpreter and web application will be completed using an Agile/Scrum methodology. Both design and development techniques were initially taught in the Software Engineering module and are widely used in the industry today.

The programming for the project will predominantly be completed using Python 3. According to GitHub's Octoverse Report, Python outranked Java as the second most popular language on GitHub by repository contributors [5]. Python is one of the most popular programming languages and is a useful skill to develop for this undergraduate course and future job opportunities. Using API's, frameworks and hosting web applications on cloud services will be useful practical skills for future job opportunities and education.

1.2 Professional Considerations

The nature of the project using social media automation results in multiple different ethical issues and considerations. The intent of the project is to allow more automation and control across social media content for brands and businesses to engage with their audience, however social media automation can be used for many other reasons and this leads to many ethical considerations and consequences. Examples of real world negative ethical consequences of social media automation are spam, harassment, spreading false information, troll armies and influencing content reach and impressions.

1.2.1 US Presidential Election 2016

One of the most infamous incidents in which Twitter and Facebook bots have been used in recent years is the role the bots took in the United States 2016 presidential election [13]. Twitter bots were claimed to be configured to interact with certain political figures. This was done to give these tweets more exposure, boost their ranking on Twitter's trending page, manipulate traffic and spread false information. This incident has been referred to as a threat to democracy and Twitter has received increasing pressure to deactivate and restrict bot accounts [11]. It is estimated that nearly 48 million Twitter accounts are operated by bots, with 15% of the total accounts coming from automated bots [15]. The use of bots to spam content and engage with certain accounts is prevalent across all social media platforms and is a growing trend.

1.2.2 BCS Code of Conduct

The project will adhere to all BCS code of conduct standards. With the nature of the project and ethical issues discussed, Section 1 Public Interest is an import standard. It is important to have regard for public health, privacy and security and I shall achieve this in the project by directly implementing API limitations and rules from the platform. This will help with public health as restrictions will be placed on the frequency of posts to reduce spam, trolling and harassment.

Another important standard is Section 2 Professional Competence and Integrity. Section 2(c) is important to my project as the system will be designed for practical application. The system will be designed to comply with all current technological developments and procedures. Section 2(d) is important as the project will be used in practice. Legislation such as GDPR will be important to the project as the wider scope of the project is to include analytics. I will comply with GDPR legislation around data processing, data handling and anonymizing data.

The project aims to involve human participants for usability testing of the system. This

would require an approved ethical review and a Ethical Compliance Form will be submitted in the near future. This is to comply with Section 3(a) to carry out professional responsibilities with due care and diligence in accordance with the relevant Authority's requirements.

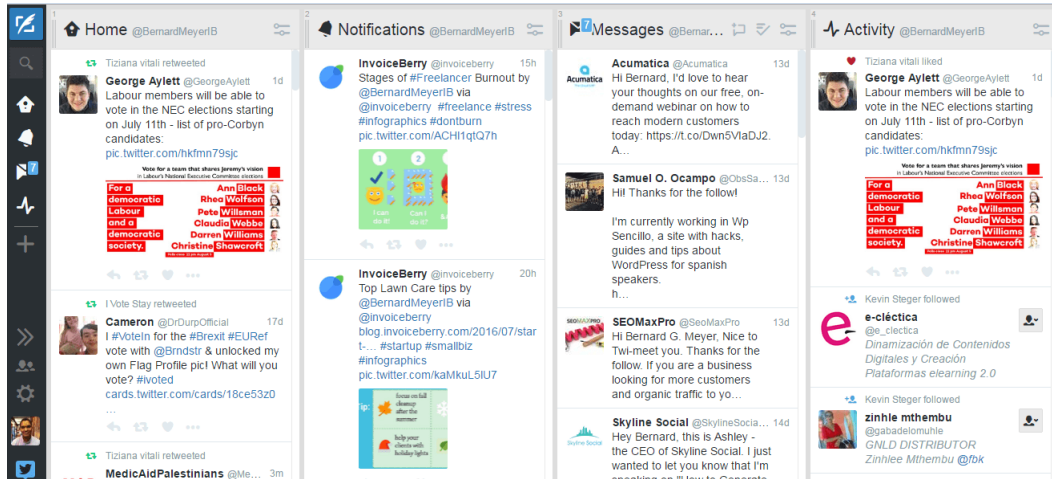
1.3 Related Work

1.3.1 TweetDeck

TweetDeck is a social media dashboard application for Twitter and is integrated into Twitter's interface. The application interfaces with the Twitter API to allow users to send and receive tweets and view profiles [3].

TweetDeck consists of a series of customisable columns which can be customised to display:

- Home: Home timeline for any specific account.
- User: Tweets from a specific account.
- Notifications: Notifications for a specific account, including when the account's Tweets are Retweeted, liked, or mentioned, and when someone follows the account.
- Search: A specific search term.
- Lists: Create or connect a list you already follow.
- Collection: A timeline of curated Tweets, hand-selected by you, to share with others.
- Activity: What's happening with the accounts you follow.
- Likes: Tweets marked as likes from a specific account.
- Messages (one account): Direct Messages for a specific account.
- Mentions (one account): When someone mentions a specific account.
- Followers: Follow activity for a specific account.
- Scheduled: Your scheduled Tweets.
- Messages (all accounts): Direct Messages from all your authorized accounts in aggregate.
- Mentions (all accounts): Mentions from all accounts.
- Trending: Specific worldwide trends.



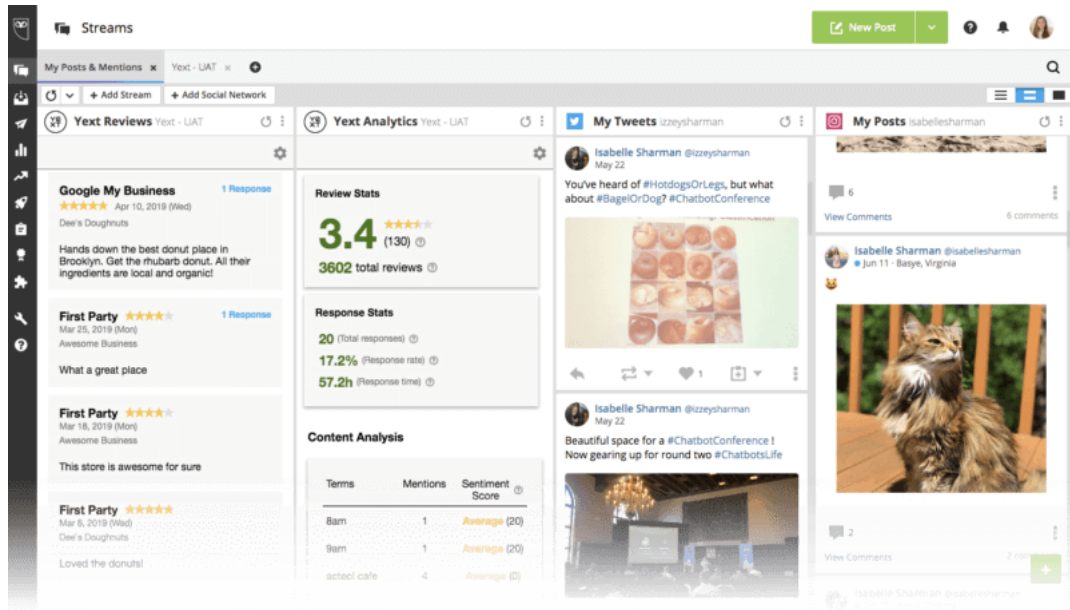
TweetDeck Dashboard [7]

TweetDeck has advanced features for consuming and analysing lots of information at once. TweetDeck supports advanced features such as Tweet Scheduling, collections, search, filtering and more [6].

- Scheduling: TweetDeck allows users to schedule Tweets in advance.
- Collections: Collections are organised Tweets according to topics, events, interests, and conversations all in real time and can be added as columns.
- Advanced search: TweetDeck runs real-time search including sentiment.
- List Management: TweetDeck allows users to manage lists in one centralised place for multiple accounts. Lists can be created by filtering interests or by particular accounts.
- Embedded Tweets: TweetDeck can create embedded Tweets directly from the app.

1.3.2 Hootsuite

Hootsuite is a social media management tool which manages over 35 popular social networks in one place. The system's user interface takes the form of a dashboard to monitor multiple streams in one place [1].



Hootsuite Dashboard [10]

Core features of the Hootsuite platform include:

- Scheduling
- Content Curation
- Promoting
- Analytics
- Monitoring
- Team Management
- Security
- Apps and Integration's

Hootsuite provides other products:

- Amplify: Allows staff to share company news
- Insights: Presents data in real-time through visual reports
- Impact: Generates ROI for marketing campaigns
- Ads: Platform for social advertisement

1.3.3 Social Bot

Social bot is a project stored in a public repository on GitHub. The project description in the README is Python bots for social networks. The project supports Twitter, Instagram & Facebook. This project provides most functionality for Twitter, as it can post, reply, retweet, quote and like tweets. It can also follow & unfollow users, get users and tweets by a term search and get user profile details. Instagram and Facebook bots are more limited for this project as they cannot post or schedule and can only retrieve user data and posts. The API provides granular action-functions that can be used by bots operating individually or as swarm [4].

1.3.4 Software Bots

Software Bots is a report published by IEEE which explores the types of bots, platforms for working with bots and offers guidelines on how to create and use them.

According to the report, bots can be categorised through their interaction model. Some bots support a domain-specific language in which users interact with bots using specific commands in a command line interface while other bots might parse natural language through text or speech. Other bots can also include UI controls that lets users respond and interact quickly. Bots can be characterised according to their purpose:

- Generalist bots such as Siri or Cortana support a range of simple tasks and direct users to appropriate external resources when deeper knowledge is required.
- Transactional bots work on the users' behalf, automatically executing transactions with external systems (for example, automatically making a purchase when a price level is reached).
- Informational bots fetch information for users (for example, gathering insights about stocks or providing weather updates).
- Productivity bots improve user or team productivity by automating rote or tedious tasks (for example, updating calendars or silencing notifications).
- Collaboration bots help users communicate, coordinate, and collaborate

There are multiple tools for bot development and these tools are distinguished for building bots (creation platforms) and the platforms on which the bots dwell (distribution platforms). Creation platforms provide a variety of software foundations, frameworks, tool-kits, APIs and other features. These provided services range from well documented code templates to out the box no-code required bot-building interfaces. Distribution platforms dictate where and how users access bots and many focus on social networking and messaging.

With the increase in popularity of bots and the significant role they play in day-to-day

life it is important to understand what works well and what does not. Users should always know what to expect and bot developers should make it clear to users that they are interacting with a bot. User-bot interaction should be a smooth and frictionless experience and bots should do no harm. Developers should carefully consider how their bots could be misused intentionally or unintentionally. Many bots are already deemed malicious so building user trust poses challenges [14].

1.3.5 Research

The background research for the related work includes two industry standard social media management tools, an online repository for social media bots and a report around the current state of software bots. There was limited background research for domain specific languages for software bots as it is not a well researched topic and social media management tools do not have open source code. Research was therefore conducted on industry tools such as TweetDeck and Hootsuite, an example of a social media bot and a published report on all aspects of Software Bots.

This research was important as TweetDeck demonstrates all the functionality of a web application solely using the Twitter API. This application is similar to the overall scope of the project however the project aims to provide a similar application across multiple platforms. Hootsuite is an all in one robust social media management tool. It provides methods of high level abstraction from interacting with the social media platforms API through a drag and drop centralised dashboard. Hootsuite is similar towards the overall goals of the project however the project aims to abstract the interaction of the API's through a domain specific language and separate the web application from this. Hootsuite includes streamlining team management and integrated analytics which is outside the scope of the project.

Social Bot demonstrates the functionality of social media bots using the API's in a command-line interface. This is important to the project as it demonstrates what can be achieved using a different approach. Software bots presents information about the different types of bots, categorising these bots, how the bots work and what platforms they work on and solutions to ethical implications bots pose. This was useful initial research as it allowed direction for the project and helped with the content of the interim report.

1.4 Requirements Analysis

The system is aimed to be directed towards general users, brands and agencies who use social media daily to connect with their audiences. It is vital to understand the needs of this audience, to what extent current solutions meet their needs and what else they would like to see from current solutions. To help understand this, an interview was conducted with InCrowd.

InCrowd is a company based in London and Brighton who provide intuitive sports technology to help brands engage with their audiences. The interviewees who work for InCrowd are Matt and Helen, who work in the Marketing team for InCrowd. Matt works with SkyBet and solely uses Twitter and native Twitter tools in his day to day job role. Helen works in the Business to Business (B2B) marketing team and focuses on using Twitter and LinkedIn to achieve this role. Both interviewees use the native platforms tools and applications but have used other industry standard social media management tools to automate and schedule content in previous job roles.

To understand InCrowds needs, functional and non-functional requirements are produced to identify their needs and provide scope on the overall direction of the project. The requirements analysis has been completed from the interview that was conducted with InCrowd. InCrowd provided ethical consent to use selected quotes from the interview in the report and this is included in the appendix. Further requirements have been extracted from related work as some features are essential for the project. The project is designed to meet InCrowds needs however it has been designed with the intention to be used by any brand or agency and not exclusively for InCrowd.

1.4.1 Functional Requirements

Reference	Description	Mandatory/Desirable
F1	The system shall provide the basic functionality of what actions can be completed by a human on Twitter which can be configured by the domain specific language. This functionality includes login, post tweets, reply to tweets, retweet tweets, like tweets and follow users.	Mandatory

F2	The system shall provide the basic functionality of what actions can be completed by a human on Instagram which can be configured by the domain specific language. This includes getting an Instagram profile and getting an Instagram user's images, video's and albums.	Mandatory
F3	The system shall provide the basic functionality of what actions can be completed by a human on Facebook which can be configured by the domain specific language. This functionality includes posts, reading notifications and messages.	Mandatory
F4	The system shall allow simultaneous scheduling for content across Twitter, Instagram and Facebook.	Mandatory
F5	The system shall provide the ability to have a different content set optimised for each social media platform in the same content schedule.	Mandatory
F6	The system should provide a web interface as a single dashboard for different feeds.	Mandatory
F7	The web interface shall be similar to Tweetdeck's UI having multiple columns.	Mandatory
F8	The system shall provide customisable columns for multiple social media platforms for F7.	Mandatory
F9	The system shall provide the functionality of personalised automated support responses for directing messaging on Twitter and Facebook.	Mandatory
F10	The system should retrieve a list of relevant content from other users on the platform based on a customisable input such as keywords and hashtags.	Desirable
F11	The system shall provide the ability to remove content from the list provided in F10 and provide the ability to interact with the content with the use of a single click.	Desirable

F13	The system should have a central messaging system which provides the functionality of directly message users across multiple platforms in one inbox.	Desirable
F14	The system should provide a list of suggested accounts to follow based on inputs such as keywords and hashtags.	Desirable
F15	The system should provide a list of people who engage with the accounts the most in a given time frame.	Desirable
F16	The system should provide the ability to customise reports within a given time-frame.	Desirable
F17	The system should be integrated into In-Crowd's native insights tool Bridge.	Desirable

1.4.2 Non-Functional Requirements

Reference	Description	Mandatory/Desirable
NF1	The domain specific language shall run on Windows, Mac and Linux Operating Systems.	Mandatory
NF2	The domain specific language shall be programmed using Python 3 and use ANTLR 4 toolkit.	Mandatory
NF3	The interpreter shall run on Windows, Mac and Linux Operating Systems.	Mandatory
NF4	The interpreter shall be programmed using Python 3.	Mandatory
NF5	The web application shall run on all latest browsers.	Mandatory
NF6	The web application shall be programmed using Python 3 and Django Web Framework.	Mandatory
NF7	The web application shall use Django's user login system to provide security for the web application through user authentication.	Mandatory

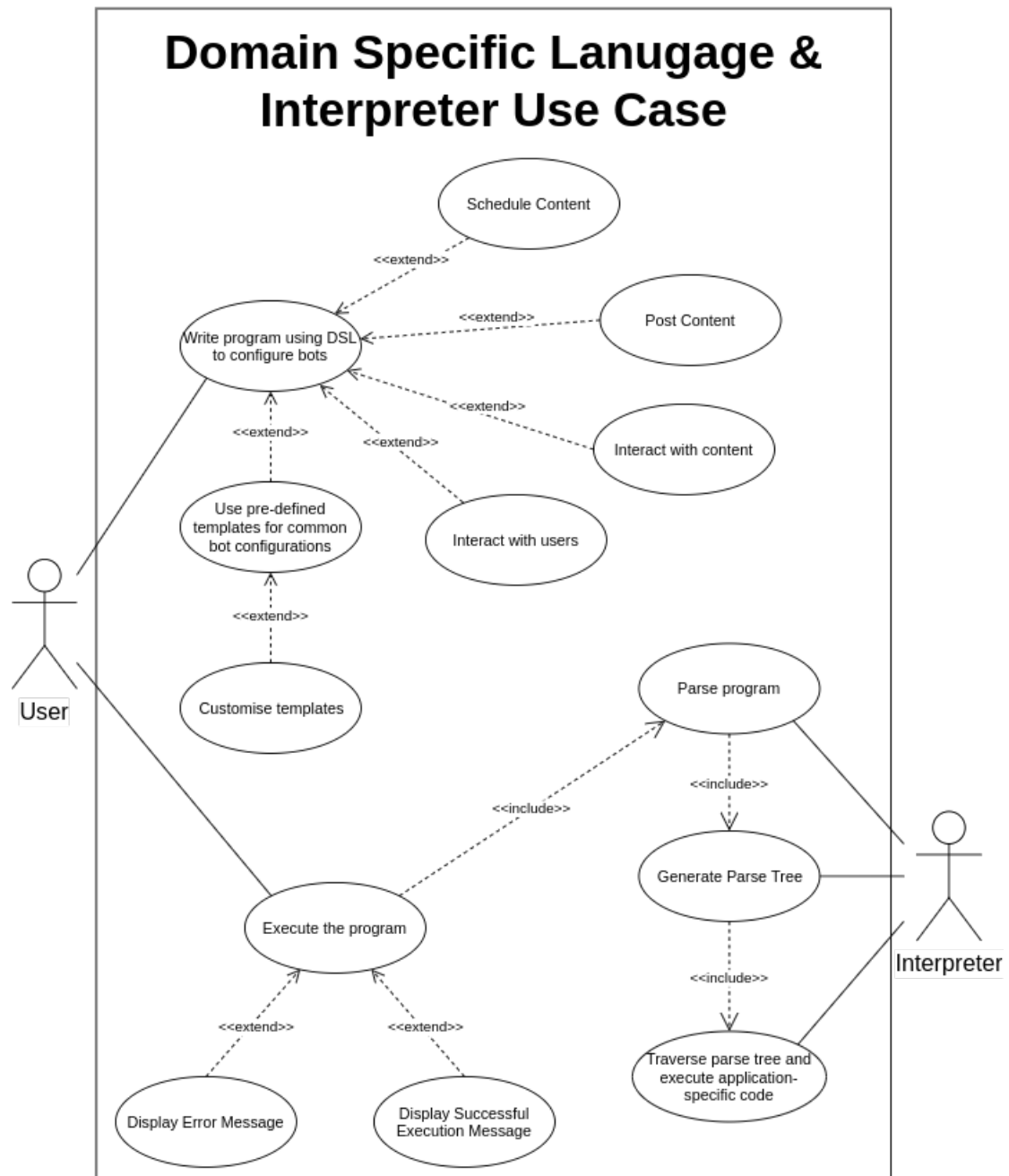
1.4.3 Requirements

The ideal system for InCrowd would be a social media management system which allows scheduling and automated content across all platforms integrated into their proprietary insights system. The project aims to contribute to their needs by providing a domain specific language to automate and schedule content and provide a customisable web interface to act as a social media management system. The project will not be able to be integrated into their proprietary insights system.

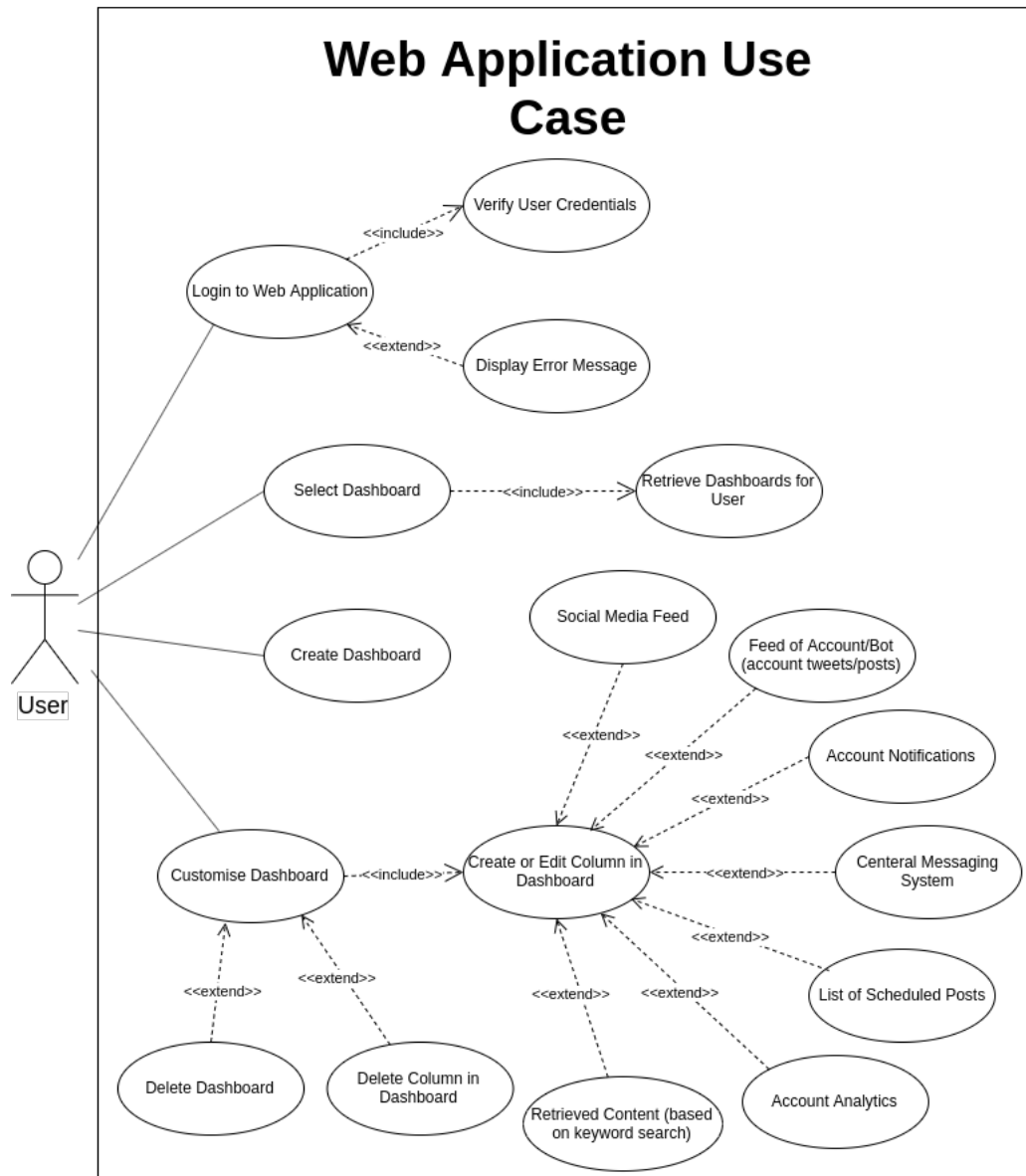
The automation and scheduling of content and web application is expected to be completed within the given time frame. The goal is to complete all mandatory requirements and as many desirable requirements as possible however some of the desirable requirements are intricate tasks which may not be completed within the time frame. The project will be designed to be scalable so additional functionality such as analytics and reported can still be integrated at a later date.

1.4.4 High Level Diagrams

UML Use Case diagrams have been created to visualise the functional requirements analysis. It demonstrates the functionality of the system and how the user will interact with the system.



UML Use Case Diagram



UML Use Case Diagram

1.5 Project Plan

The project can be split into 5 main tasks. The first main task is the interim report, the next three tasks are three major objectives of the project and the final task is the final report.

1.5.1 Task 1: Proposal and Interim Report

- By 10th October 2019, the first draft for the proposal will be completed. The proposal will be finalised with minor potential changes to the smart objectives.
- By 17th October 2019 the proposal will be finalised. This completed document will be approved by the project supervisor and will not change.
- By 25th October 2019 the meeting with InCrowd will be completed. This includes a full interview transcript and ethical approval & consent.
- By 8th November 2019 the first draft for the interim report will be completed. The interim report will be finalised with minor potential changes based on supervisor feedback.
- By 14th November 2019 the finalised document for the interim report will be completed. This document will be approved by the project supervisor and will be submitted online.

1.5.2 Task 2: Domain Specific Language

Design

- By 14th November 2019, the decisions about what languages, tools and frameworks to be used will be submitted in the interim report.
- By 14th November 2019, high level diagrams for how the user will interact with the system and how this reflects the language design will be submitted in the interim report.

Implementation

The implementation of the Domain Specific Language will be completed in an agile methodology therefore the intricate details of the implementation are not included. What is expected to be completed at deadlines of prototypes, versions and the final system are included.

- By 30th January 2020, the first version of the domain specific language will be completed. This prototype will be able to parse an input and generate parse trees based on the grammar of the language.
- By 29th February 2020, the second version of the domain specific language and the parser will be completed. This version will be refactored to work with the first version of the interpreter and will likely involve several agile sprints.
- By 10th April 2020, the final version of the domain specific language and the parser will be completed. This prototype will include the full language for all functionality of the mandatory functional requirements. This version will be expected to have minor changes and only include bug fixing agile sprints to fix any errors.
- By 1st May 2020, the final version of the domain specific language and the parser will be completed. This version will be fully functional and will be the version that is submitted in the final report.

Testing

The testing for the domain specific language will be completed using unit tests. Git Hook will be used to ensure that unit tests pass when committing to the version control system. Unit tests will be completed during each agile sprint cycle and will thoroughly test the work completed in the cycle.

- By 1st May 2020, the unit tests for the domain specific language will be completed. These unit tests will cover all functionality for the final version of the project and testing documentation will be made available in the final report.

1.5.3 Task 3: Interpreter

Design

- By 14th November 2019, the decisions about what languages, tools and frameworks to be used will be submitted in the interim report.
- By 30th January 2020, formal designs for the interpreter will be completed.

Implementation

The implementation of the interpreter will be completed in an agile methodology therefore the intricate details of the implementation are not included. What is expected to be completed at deadlines of prototypes, versions and the final system are included. The

development for the interpreter will commence on January 30th after the first prototype of the domain specific language and parser has been completed.

- By 14th February 2020 the interpreter will execute the domain specific language and provide the functionality to sign in, tweet, like, follow and retweet on Twitter.
- By 29th February 2020 the interpreter will execute the domain specific language and provide the functionality in FR1, FR2 and FR3 in the requirements analysis.
- By 10th April 2020 the interpreter will have completed all functionality for the mandatory requirements and as many desirable requirements as possible within this time frame. This version of the interpreter will be expected to have minor changes and only include bug fixing agile sprints to fix any errors.
- By 1st May 2020 the final version of the interpreter will be completed. This version will be fully functional and will be the version included in the final report.

Testing

The testing for the interpreter will be completed using unit tests. Git Hook will be used to ensure that unit tests pass when committing to the version control system. Unit tests will be completed during each agile sprint cycle and will thoroughly test the work completed in the cycle.

- By 1st May 2020, the unit tests for the interpreter will be completed. These unit tests will cover all functionality for the final version of the project and testing documentation will be made available in the final report.

1.5.4 Task 4: Web Application

Design

- The decisions about what languages, tools and frameworks to be used will be completed by 14th November in the interim report.
- By 1st March 2020 formal designs for the UI of the web application will be completed.

Implementation

The implementation of the Web Application will be completed in an agile methodology therefore the intricate details of the implementation are not included. What is expected to be completed at deadlines of prototypes, versions and the final system are included.

The development for the interpreter will commence on 1st March 2020 after the first prototype of the interpreter has met the initial functional requirements.

- By 1st April 2020, the first prototype of the web application will be completed. This will include all mandatory requirements functionality.
- By April 10th the web application will have completed all functionality for the mandatory requirements and as many desirable requirements as possible within this time frame. This version of the web application will be expected to have minor changes and only include bug fixing and UI agile sprints to fix any errors.
- By 1st May 2020 the final version of the interpreter will be completed. This version will be fully functional and will be the version included in the final report.

Testing

The testing for the web application will be completed using unit tests. Git Hook will be used to ensure that unit tests pass when committing to the version control system. Unit tests will be completed during each agile sprint cycle and will thoroughly test the work completed in the cycle.

- By 1st May 2020, the unit tests for the web application will be completed. These unit tests will cover all functionality for the final version of the project and testing documentation will be made available in the final report.

1.5.5 Task 5: Final Report

- By 24th April 2020 the first version of the report will be completed. The only changes to this report will be feedback by project supervisor.
- By 1st May 2020 the final version of the report will have been completed and this version will be submitted for the project.

1.5.6 Gantt Chart

The Gantt Chart visualises the breakdown of the tasks in a chart. The chart displays the timeline of the overall tasks to be completed and it includes milestones at the bottom to show when each version/prototype of the task should be completed by. The Gantt Chart is currently set to the date the interim report is due, with the interim report, planning and designs already completed.

The tasks are dependent on each other and this has been displayed in chronological order. The domain specific language has to be developed first, the interpreter second

and the web application can be developed at any given time. The web application will be developed last as it is not as essential to the project as the domain specific language and the interpreter and the task has a shorter time frame as it is the most competent problem area.

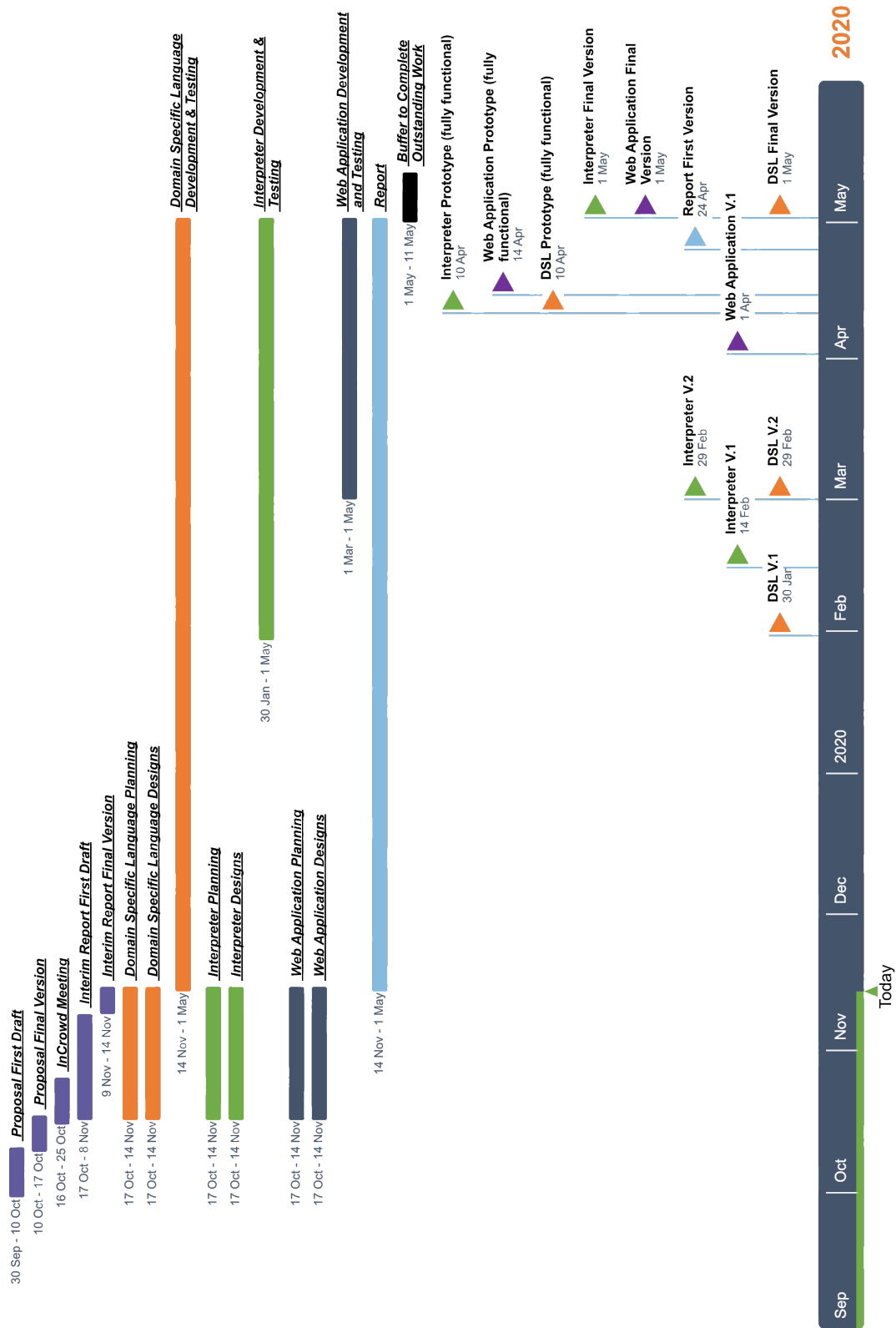


Figure 1.1: Gantt Chart

2 Appendix

2.1 Log

2.1.1 Meeting 1: 03/10/2019

Agenda

- Discuss next steps and what to do as first meeting.

Discussed Topics

- Proposal: Look at the guide of what to include in the proposal and start working on the proposal
- Research: Understand the current market of bots and their functionality and capabilities. Start to research on tools on which you would use.
- Meetings: Arrange a meeting with InCrowd sports.

Next Steps

- Work on proposal, arrange meetings and do some general research.

2.1.2 Meeting 2: 16/10/2019

Agenda

- Discuss the research I have completed to gain more scope of the project.

Discussed Topics

- The current stage of research and proposal and what needs to be improved.
- What programming languages/frameworks and packages/tool kits which might be helpful for the project.
- Recommended books - ANTLR book for DSL's

Next Steps

- Finalise a date and time to meet with InCrowd.
- Email proposal to get feedback.

2.1.3 Meeting 3: 24/10/2019

Agenda

- Discuss the proposal and what can be improved, improve and finalise the questions for InCrowd meeting and discuss the ethics around the interview.

Discussed Topics

- Proposal: Feedback on the proposal and where to improve
- Meeting: Finalise meeting questions for InCrowd
- Ethics: Ethical guidelines and review for the meeting. Will need consent forms and will need ethical approval if the system is used.

Next Steps

- Start working on the interim report
- Arrange meeting after Ian's China trip to get feedback and answer questions about the interim report.

2.1.4 Meeting 4: 11/11/2019

Agenda

- Introduction
 - How much detail should I go into? Should I aim for it to be easily understandable and then go into enough detail for the reader to understand if they were a final year undergraduate student?
 - Do I include aims and objectives in the introduction and Smart objectives?
- Professional Considerations
 - How much detail should be discussed in regards to the ethics of bots? Should examples such as the 2016 US election be discussed?
 - What do I write about the ethics approval and do I need ethical approval from InCrowd?
- Related Work
 - How many examples should I include and should I include academic papers?
- Requirements Analysis?
 - Should high-level diagrams be included?
 - Should the transcript be included in the appendix?
- Log
 - Should the log in a minutes format?
- Proposal
 - Should this be updated to reflect the current scope of the project?

Discussed Topics

- The topics discussed were the different sections to be completed for the interim report
 - Introduction: The introduction should look at the problem area and introduce your solution to the problem area. Discuss if it is possible to implement a domain specific language for the project and why a programming driven solution might work.
 - Professional and ethical considerations: Discuss Cambridge Analytica, the risks of automation, how your project could be weaponised and acknowledge

the dangers of the project and your solutions to this. You will need consent from Matt and Helen to put in the appendix and may only use selected quotes.

- Related Work: Include an academic paper and link the requirements analysis to the related work.
- Requirements Analysis: Include high level diagrams.
- Log: The log does not need to be converted into minutes format.
- Proposal: Update the proposal when necessary.

Next Steps

- Email draft to get feedback.
- Arrange meeting for early next week to start project.

2.2 Proposal

2.2.1 Goals

The goal of this project is to design and implement a domain specific language, a parser and its run time to control the interaction of social media bots with multiple interactive feeds. The DSL will contain the features and capabilities of what a user can do on the social media platforms so they can be automated and scheduled. The goal is for social media marketing teams and individuals to easily automate interactions on the social media platforms and have a web application with a customisable interactive feed for the bots account.

2.2.2 Objectives

The project can be split into three main tasks. This includes:

1. Designing and implementing the domain specific language to produce executable code/commands
 - a) The overall aims of this task are to design and implement the domain specific language. This includes designing and implementing the grammar for the language. The grammar will be used by ANTLR to generate a parser that can build and walk parse trees.
2. Implement the interpreter to configure and operate the social media bots
 - a) The overall aims of this task are for the interpreter to use the parse tree to translate it into some efficient intermediate representation and immediately execute this. The interpreter will explicitly execute stored pre-compiled code made by a compiler which is part of the interpreter system. This code will interact with the social media API's to configure and operate the bots.
3. Implement an interactive web interface which displays the feeds/interactions for that bot/account
 - a) The overall aims of this task are to create a web application that displays customisable feeds for the social media account that a user can interact with. The web application will require security, multiple user capabilities and web-hooks for the social media API's to create a personalised feed for the individual user. The interface will be fully customised, which can display all the different feeds for multiple social media accounts on one web application.

SMART Objectives for research, planning and interim report

- By October 21st 2019, complete a research document to help with the scope of the project, to understand its potential and identify similar industry technologies.
- By October 25th 2019, a meeting will have taken place with InCrowd to discuss their business model, what they would like to see from the project and the current state of the technology and gaps in the market.
- By November 14th 2019, the interim report will include GANTT charts, final planning documentation and final designs using high level and low level designs and modelling.

SMART Objectives for Task 1

- By November 14th 2019, all designs for the language will have been completed and development will begin.
- By January 30th 2020, a first prototype for the domain specific language will be completed.

SMART Objectives for Task 2

- By November 14th 2019, all designs for process of configuring and operating the bot will be completed.
- By February 25th 2020, the domain specific language will be able to run and execute commands for social media bots.
- By March 9th 2020, the social media bot including the DSL will be completed and finalised. After this point there will no longer be any development dedicated to the Bots.

SMART Objectives for Task 3

- After March 9th 2020, development for the interface and feeds will begin in an agile methodology.
- By April 13th 2020, the first version of the project will be finalised.

2.2.3 Relevance

The relevance of this project extends to various topics that have been studied at University and covered in previous work experience.

The design and implementation of the domain specific language, parser and interpreter was covered in the Compilers and Computer Architecture module.

The domain specific language will require rigorous design to ensure that it is scalable, efficient and usable and will require more traditional waterfall planning and techniques. The interface used to display the feeds and interactions of the bots will be developed using Django, which will require an Agile/Scrum methodology. Both these design and development techniques and methodologies are relevant as they were covered in the Software Engineering module, have been used with previous work experience and are widely used in the industry today.

The programming will be predominately done using Python 3, which I have learnt through the Natural Language Engineering module and I have used in industry. I have also used the Django framework written in Python.

I have currently not used any API's in University however I have used some Amazon API's with previous work experience.

The primary use of the project is for social marketing teams and individuals to have the ability to automate social media posts and interactions with the posts. I wanted to create something practical and usable and this was all inspired by previous work experience working with paid search marketing teams and tools. The relevance of this project links to previous experience I have and potentially the industry that I would like to work in post University.

2.2.4 Resources Required

No resources required other than access to programming languages, APIs, IDEs, frameworks and packages. All these resources are open source, free or available to use on Software Hub. The resources I that I wish to use are Python 3, ANTLR, Twitter API, Facebook API, Instagram API, Git and Github, Django (Python Web Framework), L^AT_EX, Clubhouse and IntelliJ. I plan to do all development on my laptop running Ubuntu 18.04 LTS.

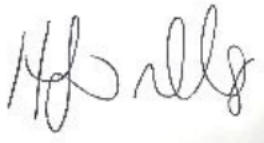
2.3 Ethical Consent

I approve that Samuel Withey may only use selected quotes from the transcribed interview that took place on 25/10/2019 in all aspects of his final year project.

Matt Galliford

A handwritten signature in black ink that reads "matt galliford" in a cursive, lowercase style.

Helen Nicholls

A handwritten signature in black ink that reads "Helen Nicholls" in a cursive, uppercase style.

Helen Nicholls, Marketing Executive - InCrowd

Bibliography

- [1] Social media management dashboard - hootsuite.
- [2] Hootsuite global digital report, 2019.
- [3] How to use tweetdeck, 2019.
- [4] Python bots for social networks, 2019.
- [5] The state of the ovtoverse, 2019.
- [6] Tweetdeck advanced features, 2019.
- [7] Tweetdeck dashboard, 2019.
- [8] M. Bravenboer, Éric Tanter, and Eelco Visser. Declarative, formal, and extensible syntax definition for aspectj - a case for scannerless generalized-lr parsing. *ACM SIGPLAN Notices*, 41, 10 2006.
- [9] Martin Fowler. *Domain-Specific Languages*. The Addison-Wesley signature series. Addison-Wesley, 2011.
- [10] Amy Gesenhues and Amy Gesenhues. Hootsuite dashboard, 2019.
- [11] Marina Gorbis and Samuel Woolley. Social media bots threaten democracy., 2019.
- [12] Dick Grune, Kees Van Reeuwijk, Henri E Bal, Criel JH Jacobs, and Koen Langendoen. *Modern compiler design*. Springer Science & Business Media, 2012.
- [13] Philip N. Howard, Samuel Woolley, and Ryan Calo. Algorithms, bots, and political communication in the us 2016 election: The challenge of automated political communication for election law and administration. *Journal of Information Technology & Politics*, 15(2):81–93, 2018.
- [14] C. Lebeuf, M. Storey, and A. Zagalsky. Software bots. *IEEE Software*, 35(1):18–23, January 2018.
- [15] Onur Varol, Emilio Ferrara, Clayton A Davis, Filippo Menczer, and Alessandro Flammini. Online human-bot interactions: Detection, estimation, and characteriza-

tion. In *Eleventh international AAAI conference on web and social media*, 2017.

- [16] Markus Völter, Sebastian Benz, Christian Dietrich, Birgit Engemann, Mats He-
lander, Lennart C. L. Kats, Eelco Visser, and Guido Wachsmuth. *DSL Engineering
- Designing, Implementing and Using Domain-Specific Languages*. dslbook.org, 2013.