

Bringing Knowledge Through AI and SMS

Sam Heather
Department of Computer Science,
The University of York,
`sam@heather.sh`

February 15, 2015

Abstract

In remote Africa there are millions of disadvantaged and uneducated individuals, who in the vast majority do not have access to the internet the knowledge that this brings. Outside of their immediate friends and family, individuals can not access the information they need to further their education, or just for general interest.

In many parts of the world, the number of people without access to the internet but access to a mobile phone is significant. This project aims to research and develop a system capable of bringing knowledge through a question and answer based interactive system, in the language natively spoken by the user, through the use of a simple Artificial Intelligence and an SMS interface. The system will be expandable, such that it can be adjusted to handle questions on any knowledge area.

This project raises ethical issues relating to the responsibility of providing accurate information when in a position of trust, the ethics of machine translation and maintaining user privacy.

It is early days so it is normal that many things are missing from the abstract. You should keep in mind that it should include later on few sentences on method/experiment, result and conclusion.

Contents

1	Useful thoughts from Lilian	9
2	Introduction	10
2.1	Background of this project	10
2.2	Motivation for this project	11
2.3	Aims of this project	11
2.4	Structure of this report	12
2.5	Assumptions Within This Project	13
3	Literature Review	14
3.1	Previous work using SMS	14
3.2	Ethical Issues	15
3.2.1	Ethics of Providing Information	15
3.2.2	Ethics of Translation	16
3.2.3	User Privacy and Data Protection	17
3.3	Software Design Life Cycles	19
3.3.1	The Waterfall Model	20
3.3.2	The Iterative Model	21
3.3.3	Boehm's Spiral Model	22
3.3.4	Agile Development	22
3.3.5	Development Life Cycle Summary	23
4	Method and Requirements	24
4.1	Plan for Software Development	24
4.2	Requirements	24
4.2.1	Use Cases / User Goals	24
4.2.2	User Requirements	24
4.2.3	System Requirements - Functional	25
4.2.4	System Requirements - Non-Functional	25
4.3	Evaluating Success	26

5 Design	27
5.1 Software Design	27
5.1.1 Micro-system for Protecting User Identity	27
5.2 Language, Platform and Tools and Libraries	27
5.2.1 Language	27
5.2.2 Platform	28
5.2.3 Database	28
5.2.4 Libraries	31
6 Implementation	33
7 Results	34
8 Discussion	35
9 Evaluation	36
10 Extending this project	37
11 Conclusion	38
12 Appendix - Personas	41
12.1 Nanjala	41
12.1.1 Demographics	41
12.1.2 Average Day	41
12.2 Cedric	42
12.2.1 Demographics	42
12.2.2 Average European Trip	42

List of Figures

1	Screenshots of Shy iOS App	10
2	Google Now information box. Screenshot taken on 3rd February 2015.	14
3	Hashing a phone number using an imaginary hashing algorithm	18
4	Waterfall Development Model	20
5	Iterative Development Method	21
6	Boehm's Spiral Model	22
7	Conceptual Model of Data	29
8	Logical Model of Data	30

List of Tables

1	Example of software database: hashed phone numbers paired with a list of previous returned answer ID's.	18
2	Logical Database Design. <u>Underlined</u> fields represent primary keys. . .	30
3	MySQL & SQLAlchemy vs MongoDB	31
4	Typical Document in each MongoDB Collection	32

Statement of Ethics

Pretty much the most important part - write this!

Acknowledgement

The author would like to acknowledge the following individuals for their contribution to this project:

- Julie Markham and Nicholas Hopper, with whom the author collaborated with on Shy, the mobile application that inspired this project.
- Lilian Blot, for his time and effort spent supervising and providing guidance for this project.

1 Useful thoughts from Lilian

Take a quick look at list of other students previous projects to get an idea of how intro/lit review is structured etc.

Remove page number on title page.

Remember to take notes on when I come across problems, for my conclusion.

2 Introduction

2.1 Background of this project

The inspiration for this project originates from a project the author undertook in September 2014 whilst attending Yacht Hack, a week-long hackathon on a Yacht in Croatia¹. The author co-created a project to prototype a mobile application that would facilitate the immediate answering of questions that fall within certain categories using a knowledge base, simple machine learning and profiling of users based on their usage history.

One of the initial goals of Shy was to bring knowledge via m-learning (learning using tools available on a mobile device) as opposed to e-learning (learning conducted online, usually with the help of a computer). This is because m-learning systems are available to a much larger demographic of people, because the technology (mobile devices, commonly phones) is becoming ever cheaper and more accessible.

A semi-functional prototype was completed for iOS, as seen in Figure 1. The front-end was complete, however in the backend question-answer matching was evaluated without knowledge of previous material the user had viewed. This meant that explicitly targeted answers for a question and suggestions of questions that a given user profile might be interested in could not be made. Up until this point, the service was restricted to working on iOS smart devices only, with poor quality question-answer matching. In the initial project, the knowledge base was built for questions on personal health, sex, relationships and family.

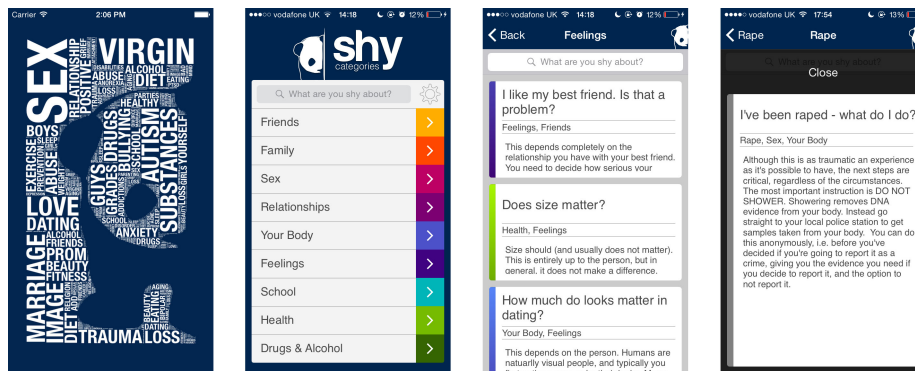


Figure 1: Screenshots of Shy iOS App

¹<http://toughhackers.com/yacht-hack-2014/>

should add at least one more significant paragraph here, not sure what to write about though

2.2 Motivation for this project

Access to knowledge is a critical part of modern life. It's also a Human Right, under Article 27 of the Universal Declaration of Human Rights [1]. We have the facility to retrieve information and instantly communicate with each other using our smart phones, and commonly do it up to 200 times a day [2] - a facility that hundreds of millions of individuals live without. This project aims to investigate the technical and ethical challenges behind this and to create a technology to give more people access to knowledge, and thus their human rights.

The project involves the use of a number of systems to work including machine translation, an SMS input/output system and a system to build a profile on a user in order to facilitate high-quality targeted question-answer matching. The system will also feature machine learning so that as time progresses, the quality of the answers it returns increases. The use of these systems in one application along with complex ethics and privacy issues create an interesting project that draws together many technologies and discussions in a way that can be used as a basis for further research in the future.

2.3 Aims of this project

The first goal of this project is to assess and address the ethical issues that arise from the software that this project aims to create. These include the responsibility the software (and hence the software author) has, because of its position of trust, to provide accurate information and to protect the privacy of its users by storing only necessary information, among others. This project will do this by researching ethical issues relating to the technologies that the project will use, for example machine translation. This will then be used to feed the design process of the software and to specify the expected use-case of it. Finally, the resulting software will be evaluated through experimentation, with volunteers asking the software a set of questions within a topic area, in a non-English language, and evaluate the relevance of answers returned.

The initial goal for this dissertation was to focus on questions relating to personal health, sex, relationships and family, which was as a direct result of the focus of 'Shy'. Although a health and diseases database was located in the early stages of the project (Diseases Database, www.diseasesdatabase.com), it was targeted at professional uses for the data and came with a significant licence fee, as well as complex documentation that the provider insisted on customising specifically for each use-case. Attempts were made to acquire a free academic research licence via email, but these were not responded to. Additionally, as the project progressed in the early stages it became clear that the author's original two-page document discussing ethical considerations with the project did not cover all possible situations. Indeed, the Author and Supervisor were still discovering further significant ethical issues as far as eight weeks into the project, which would have required a complex ethics panel review, critically delaying this project. Two example ethical issues that occurred were:

- Any application has that dispenses health related information that a user might

act upon in a life changing manner has to respond with correct information, since an incorrect response could lead to a user taking an action which leads to them harming themselves.

- Privacy is critical since the questions been asked are personal. This could not be absolutely guaranteed, since a third-party service had to be used to send SMSs and, although unlikely, it was possible that this service could build profiles on users and identify them.

As a result of the above issues with locating a reputable knowledge base and the complex ethical issues, a decision was made in early December 2014 to switch the target topic for questions to Geography and to use Wikipedia as a datasource, since it is openly available, well populated and multiple methods exist for extracting information from it. In future work, it should be possible to extend this project to work with medical questions, if a high quality data-source is located and a solution to the ethical issues is found.

2.4 Structure of this report

This report starts with a review of pre-existing literature on this topic in chapter 3, where the author looks at existing software, tools and services of a similar type to those that will be used in this project. Comparisons between different software development life-cycles are also discussed.

Chapter 4 describes the method that will be taken to develop the software and service. This covers the software development life-cycle that will be followed, and sets out a plan for when development will take place. Requirements will also be identified, described and categorised as either functional or non-functional. A method of evaluating the success of the software will also be discussed and chosen.

The design of the software, driven from information collected from the requirements, will be set out in Chapter 5. The individual components that make up the technology will be described and discussed, followed by a description of how these modules will interface with each other to build a complete system. Finally, this section will include a description of the platform, language and other tools that will be used.

described and
discussed, or
just described?

Chapter 6 contains details about the implementation of the software. This includes information on the tools that were used to create components such as the SMS interface and question-answer matching system.

Results from the evaluation of the system will be presented in Chapter 7. A discussion of these results and their implication on this project is then presented to the reader in Chapter 8.

Chapter 9 contains an evaluation of the software produced and the process taken to complete it. Comparisons will be made to the success criteria, set out for the project in section ;insert number here;.

The author's thoughts on potential future work, building on the work produced in this project, are then displayed in chapter 10.

Finally, the main achievements of this project and then the report is concluded in chapter 11 with final thoughts.

2.5 Assumptions Within This Project

This project will make the assumption that users of this service have basic literacy skills in a language supported by the project. Although this is not a correct assumption world-wide, expanding the remit of this project to include a 'graphical' user interface that works over SMS would scale this project outside of the acceptable/expected range for an undergraduate dissertation.

The main use-case for this project is going to be parts of the world with limited internet access. Although these exist all over the world, the African continent is typical of this problem so I will focus my research in demonstrating a need for this tool there.

extend this

3 Literature Review

The point of this is to show that I 'own' the material and subject (competence). Show different points of view, choose a side, express a view on which I prefer and if I agree, critique. Have objective goals.

A significant amount of work has previously been undertaken in the area of making knowledge quickly accessible to us in the form of questions and answers. To take one example, consider that an individual needs to find the population of London as quickly as possible. The initial step would be to search for information. In the connected world, this is easy: a simple Google search returns the answers without even having to click through to any resources, as shown in Figure 2.



Figure 2: Google Now information box. Screenshot taken on 3rd February 2015.

This is easy to do in the developed world where up to 75% of the population are connected to the internet [3], with the population of elderly people been largely responsible for the remaining 25% [4]. This contrasts strongly with the situation in the African continent, where in 2013, internet usage had only reached 16% [3], a figure largely inflated by South Africa where 5% of the African population generate 2/3 of internet traffic from the African Continent [3].

3.1 Previous work using SMS

Although the above suggests that the African continent is majoritively disconnected, this is not the case. Because of restrictions in the electricity available, the ways in which a non-smart phone can be used in Africa have far surpassed those in the developed

cite that these
are lower power
consumption, or
assume?

world [5], in part due to the low power consumption of a mobile phone. Interesting such examples include automated services which SMS HIV/AIDS sufferers, reminding them of medication to take and providing access to current market information for agriculture and farming, saving farmers from making daily trips (often many kilometres) or relying on out-of-date information from a weekly radio broadcast [6].

Order ok?

Interactive systems have also been developed to operate over SMS. One successful example is mobile money platforms. These allow for users from any background to pay and be paid for goods and to transfer money across long distances at negligible cost [6]. One of the most highly adopted services is M-Pesa, which in Kenya alone was responsible for £5.7 Billion in transfers in 2012². M-Pesa gives users a balance linked to a national ID number from which they can pay for goods by sending an SMS with a cashier (recipient) number or pay outstanding bills in a similar way. Non-subscribers can also use the system by depositing money with a M-Pesa cashier in exchange for an access code, which can be sent via SMS to a contact who can subsequently redeem it with their local cashier [6].

This difference in standard use of cell phones is demonstrated in the International Telecommunications Union's 2013 report [3], which shows that in Europe, for 790 million mobile subscriptions, 53% of subscribers have mobile internet access (422 million), compared to 17% in Africa (93 million have mobile broadband, out of 545 mobile subscribers). This is due to the prohibitively high cost of accessing data services, regardless of the hardware that the user has. In 2012 in Europe, 500 MB of data per month for 12 months cost 1.2% of the average Gross National Income Per Capita (GNI pc). In Africa, the average price was 30x this, at 36.2% of an individual's GNI pc [7].

Should all this be used in motivation/justification for project?

3.2 Ethical Issues

This project raises a number of ethical issues related to translation accuracy, providing information to people in an ethically acceptable way (with focus on information accuracy) and maintaining user privacy.

3.2.1 Ethics of Providing Information

One significant issue for this project stems from providing information that may affect an individual or lead them to take a harmful action.

In a similar way to that which a teacher has a responsibility to teach accurate information to a pupil due to their position of trust, any service relied upon by a user must equally provide accurate information. The result of not doing this could be providing inaccurate information that leads to a user carrying out an action that causes harm.

²Data from Safaricom, M-Pesa operator in Kenya - actual value 817,085,000,000 Kenyan Shillings, converted to GBP on 1st November 2014 at rate of approximately 0.007.

this needs to be waaaayyyy expanded, but I struggled to find information that was useful. There's lots on remote education, and on basic ethics of teaching, but I struggled to find anything specifically relating to ethics of ensuring information you provide is correct.

3.2.2 Ethics of Translation

A significant part of this project is represented by the support of multiple languages. It is clear that translation on demand, at scale, needs to be automated by some kind of machine or algorithmic translation.

This project involves two blocks of translation. These are:

- The translation of the user's input into the language of the system (English)
- The translation of the answer to a user's question from the system language to their local language.

The second of these raises some issues, but to understand these it is first necessary to understand the two categories of machine translation.

Rule-based machine translation effectively treats human language in a similar style to programming languages. Formal grammars and lexicons are used to represent words that exist in either the source or target language, structures representing the translation of individual words or groups of words. Map structures contain mappings between individual or groups of words to their translated counterparts, sometimes with multiple results (a 1-to-many map), from which rules decide which is selected. Maps and rule sets are created by trained computational linguists [8].

More commonly, Statistical Machine Translation (SMT) is used, for example, by Google and Microsoft in their translation products [8, 9]. SMT learns mappings between strings of words of potentially unequal lengths from pre-existing original texts and their trusted human translations. The accuracy and breadth of language support for translation increases as more source material is analysed by the system, as potentially erroneous or low quality translations can be identified and flagged. SMT is also dependent on the quality of the human translation on the input material [8].

A significant issue that is present in statistical machine learning systems comes from the knowledge we assume an individual has and derives from a word; in this project, this affects the translation of the answers returned to a user. In human translation this is solved by the translator's knowledge of the difference in material culture, allowing them to append necessary information to the resulting translation that the recipient might find useful. In statistical machine learning, cultural awareness of material knowledge is a separate problem on its own. **formal reference to this, probably again from Kenny, but should be able to find better.** To take one example from Melby (2006):

”when translating a French menu, a human translator might stop to think that an English speaker in France would appreciate being told that a steak tartare is served entirely raw, even if this information is not contained in the original text (because French people might be assumed to know this already). Such a translator would be aware of differences in material culture, and would be able to empathise with the English speaker who might choose to avoid the dish, given more information” [8, 10]

This issue is a result of the translation engine not being capable of take and use a complete representation of the expected cultural differences between those who speak the input language and those who speak the output language [10].

below should really be expanded and be in the discussion. Explain the decision to keep translated answers in the database, which are pre-approved, to ensure no cultural misunderstandings.

The potential uses of this system include distributing information that may be used by individuals making important decisions, regardless of whether that is within this project or in further work based off of this project, as cultural confusion such as this could have potentially catastrophic effects.

Lilian: quote taken from [8], but quote is interpreted from [10] - so I think referencing both is correct, but for this reason rather than last reason we discussed?

3.2.3 User Privacy and Data Protection

This project uses a simple Artificial Intelligence (AI) to keep a record of information that has already been returned to a user, to attempt to save sending repetitive information (saving the cost of additional SMS). This raises another ethical issue though of privacy. The material that a user has researched is likely to be sensitive; such is the nature of health related questions. This means that the information on a user has to be kept securely, in a way that an individual user can not be identified.

One way of doing this is using a technology called hashing. Hashing is the technique of taking data as an input and generating an output value (called a 'hash') that is unique to an input. Hash functions are ideally one way functions where given input data X, the output hash will always be Y, whenever and however the hashing algorithm is executed. This allows a user's identity (phone number) to be obscured, in such a way that the mobile number can not be recovered from the database. When a question is received, the phone number (represented by variable X in the above example) can be hashed and the data for this number looked up from the database without the database application ever being aware of the phone number of the user. In the example in Figure 3, a phone number (X) is used as input to an imaginary hashing function and the output hash (Y) is shown on the right hand side. As just described, this number can then be used as the key for the user information database, shown in Table 1 to retrieve the information needed for the AI.

reference the above from security engineering ross anderson, chapter 10 and 21?

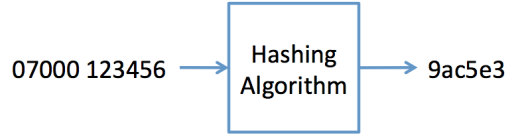


Figure 3: Hashing a phone number using an imaginary hashing algorithm

userId	userData
8f64B2	{previousQuestions:[13,301,170,577]}
9ac5e3	{previousQuestions:[441,56]}
f9dd7e	{previousQuestions:[301,623,89,280,364,621,209]}

Table 1: Example of software database: hashed phone numbers paired with a list of previous returned answer ID's.

In practise it is possible for the output of a hashing algorithm not to be unique to a particular input. This is because the range of possible input values is much smaller than the range of output hashes, and because all valid inputs must map to an output, there is repetition in outputs [11]. This property of hashing algorithms can be a security risk for some use-cases, where hashes are used to verify that data has not changed since been hashed. In this case, a third-party may be attempting to generate alternative, malicious data with the same hash as some pre-existing data, to swap them unnoticed [12]. In the use-case of this project however, hashes are only been used to anonymise phone numbers to some unidentifiable string of characters and numbers, meaning that the above-described collision risk is not a security risk.

Two properties that this project does need from its hashing algorithm are collision resistance and irreversibility. Collision resistance is the property of a hashing algorithm that determines the probability of the same output hash for two distinct random inputs [11]. A high collision resistance is necessary to ensure that two users of the system with distinct phone numbers don't get mapped to the same row in the database, resulting in poor quality question-answer matching.

Irreversibility is a necessary property of the hashing algorithm chosen for this project. This ensures that the key used in the database cannot be used to retrieve the phone number of the user, thus protecting their identity. Commonly used hashing algorithm families, such as the MD family (e.g. MD5) and SHA family are all designed to be one-way [13]³.

³Bruce Schneier is a fellow at Harvard's Berkman center for internet and society. He has been posting regularly for his newsletter and then blog since 1998 (over 16 years) and has published material related to this field throughout this time. He has been involved in the creation of other cryptographic algorithms, such as the Skein hashing algorithm, blowfish block cipher, so may have a vested interest in dis-crediting other algorithms such as MD5 and SHA. However, the essay cited above is written as a result of the Crypto 2004 Conference in California, Schneier has a world-renowned reputation for cryptography and computer security

Although hashing algorithms are computationally one-way, they can suffer from another type of vulnerability affecting the security of the original input data. When hashes are used for short strings of information (passwords, for example), a simple way to try and retrieve the original data is to compute a dictionary of the hashes of lots of common passwords, from which matches can be found. This idea has been extended to produce Precomputed Hash Chains and subsequently Rainbow Tables. These are highly efficient data structures that consist of chains of processed input messages and their hashes (with a reduce function to reach the next item in the chain), allowing original input data to be looked-up from its hash. Both Precomputed Hash Chains and Rainbow Tables tend to be Terabytes in size, and as such they only exist in a significant form for the most common hash functions (including MD5 and SHA) [14]. Within this project, this raises privacy implications, as hashes within the database would be 'convertible' to phone numbers.

perhaps the below should move to method and requirements - TODO needs to be cleaned up as well.

In modern password authentication, a tool called bcrypt is used, which 'salts' passwords (additional unique data is added to each password before hashing), nullifying the use of Precomputed Hash Chains and subsequently Rainbow Tables. However, this would prevent efficient lookup of phone numbers in the communications part of the application (where answers and sent back to the user). As such, the best way of protecting a user's phone number in the event of a database hack is to use a combination of: a largely uncommon hashing algorithm, on the basis that it is unlikely anybody will have used vast computation power to create a table that's useful in such a small number of situations; a hashing algorithm with a large output size, again because it makes a table more infeasible (because of size constraints); and a further change to the output hash, specific to this application. This way, a hacker would need to know this change (from the source code) as well as the password database, to be able to make use of a hashing table. The above is commonly implemented by hashing the result of one hash with another hashing algorithm.

3.3 Software Design Life Cycles

There are many software development life cycles that a developer or company has available to choose from. Four common ones include the Waterfall Model, Iterative Model, Spiral Model and the Agile Development Model. In order to choose which life cycle suits this project best, research into all four has been carried out.

and the algorithm that he suggests should be used at the end is not one that he has involvement in. For these reasons, I believe this source is reliable.

Did I completely forget about the existence of salt-ing?!?!?!?!?!?

Do I need to reference that ~~Should this be in this section,~~ or moved to the "Method and Requirements" section?

3.3.1 The Waterfall Model

The waterfall model consists of 5 phases of work, where each leads directly into the next. It is a plan driven development model, which means that all process activities such as the implementation of the project must be planned and scheduled before work on them begins [15]. The five layers are shown in Figure 4 and described below:

1. **Requirements.** In this stage, both user and system requirements for the project are determined after examining the business goals for the project. These form the system specification.
2. **Design.** Here, a plan for the software development is created. The overall architecture for the software is created, the external libraries to be used are chosen and the interaction between different modules is modelled.
3. **Implementation.** In this stage, the software is implemented to match the design and architecture set out in the Design phase. Unit tests are written to ensure that each module matches its specification.
4. **Testing.** In this section the system is tested as a whole to check for any bugs that might not be picked up when testing individual modules in isolation. The software is then delivered.
5. **Maintenance.** Finally, the software is deployed and maintained. This includes making changes to the requirements and subsequently the software (implementation) as the business goals change. [15]

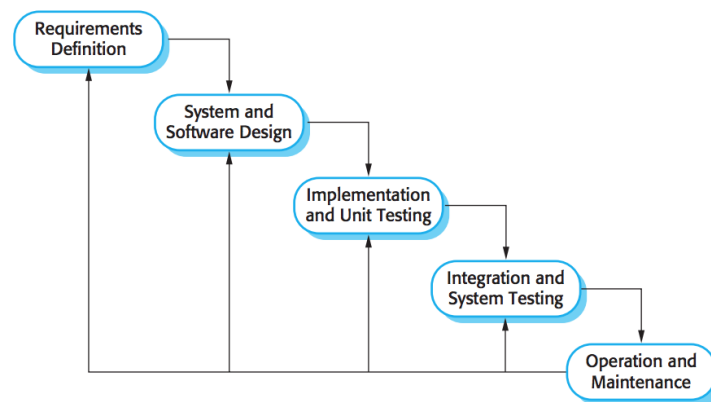


Figure 4: Waterfall Development Model
source: Sommerville [15]

A key disadvantage of the waterfall model is that once development has moved from one stage to another it cannot go back because the model is fixed and progresses

in one direction only, until the maintenance step. The main advantage of the Waterfall Model is its simplicity: it is easy to understand and to use, saving time at the start of the project that could be wasted on learning and preparing for a more complex model.

3.3.2 The Iterative Model

Iterative development is the practise of splitting the overall development of a project into multiple independent and distinct blocks. A block contains a sequence of tasks (essentially a mini life cycle) such that each iteration can really be thought of as its own project. Each iteration tends to focus on a specific set of features such that when the iteration is complete the overall project is stable and the modules developed to-date can be tested as a whole system [16].

There are a multiple variations of the iterative development method, including Timeboxed Iterative Development, where all iterations have a fixed inflexible duration, and Client-driven Iterative Development, where each iteration contains the features that the client considers have the highest business value [17].

A common variation is based on the waterfall model and is shown in Figure 5. The first three tasks in the cycle represent the first four stages in the waterfall model (Requirements, Design, Implementation and Testing), at which point there is the option to either evaluate progress so far, and start another iteration, or to deploy the software and end development. One significant difference between this and the waterfall model is that maintenance is not represented.

A key advantage of the Iterative Model is that it allows for the user to revise their requirements as they progress, making it dynamic and flexible. This is also its disadvantage, in that the lack of fixed structure can be a weakness unless the project is planned well.

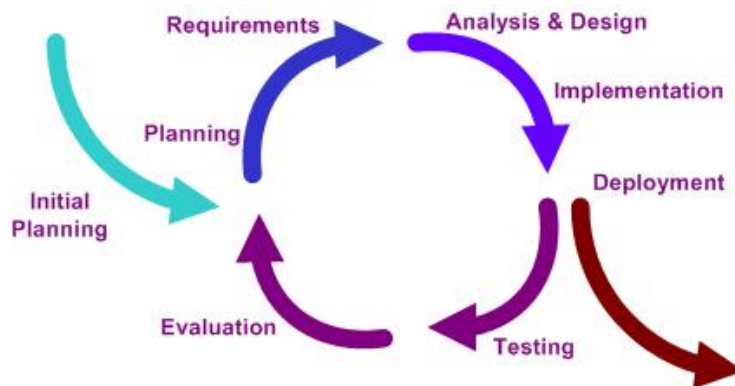


Figure 5: Iterative Development Method

source: https://commons.wikimedia.org/wiki/File:Iterative_development_model_V2.jpg

3.3.3 Boehm's Spiral Model

The Spiral Model was originally developed as a result of the adjustments that were commonly made to the waterfall model in large government projects [18]. The Spiral Model is a general model and can be used as a generator for other more specific models, given the parameters for a project such as its risks [19]. Models such as the waterfall model are specialisations of the spiral model [19].

Each loop represents a stage of the software development, from planning on the inside loop to system testing and verification on the outside, as shown in Figure 6.

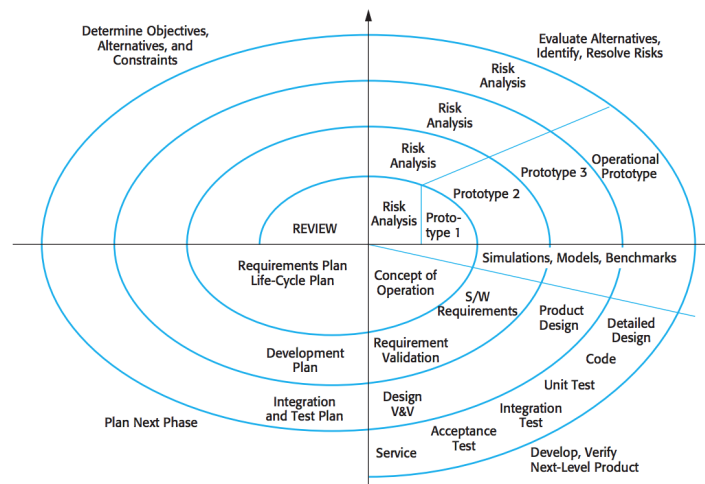


Figure 6: Boehm's Spiral Model
source: Sommerville [15]

A key advantage of the Spiral Model is the level of detail it models. The stage of a sub-task within the entire development process is represented by the distance from the origin, and each loop is split into small, well described sections. A key disadvantage is that the planning and use of the model is largely driven by the results of complex risk analysis on the sub-tasks [15]. Although this makes it favourable to large organisations, it has the opposite effect for individuals.

A detailed description on the spiral model can be found in Boehm's paper "A spiral model of software development and enhancement", where he published the complete concept [18].

3.3.4 Agile Development

Agile Development essentially represents a group of development techniques targeted towards either the development of a small to medium sized product for sale or the

development of a custom system within an organisation, where the customer has a high degree of involvement (for example 'Extreme Programming') [15].

Agile development is highly targeted towards teams, with components such as Pair Programming, Collective Ownership, regular meetings and employee health (avoiding overtime) playing an integral part in the model [15].

3.3.5 Development Life Cycle Summary

Iterative development has been chosen as the model that this project will follow, based on the following factors:

1. Boehm's Spiral Model and Agile Development are both targeted towards development teams. A significant part of these models relates to team interaction, so they are not appropriate to this project.
2. The Waterfall Model is rigid and inflexible. It does not allow for the changes to the requirements and implementation that might be proposed after evaluating the quality of the first implementation.
3. The Iterative development model can be implemented effectively by a single developer, and supports the adjustment of requirements and implementation in each iteration of the cycle.

I need to add this as a 'pro' above

4 Method and Requirements

4.1 Plan for Software Development

The Iterative Development Model was selected in Section 3.3.5. Therefore, the development of this service will follow the principles set forth in this process.

4.2 Requirements

4.2.1 Use Cases / User Goals

Two personas were created in Appendix Section 12 to represent individuals who may have a need for the work resulting from this project. From these personas, user goals are extracted. These describe what a general user aims to achieve with this service.

1. Nanjala wants to be able to read general information on any given topic.
2. Ethan wants to be able to retrieve facts from direct questions.
3. Both Nanjala and Ethan want to be able to achieve their goals entirely via SMS because of their circumstances.
4. Ethan needs a simple and efficient user interface that allows him to retrieve facts quickly.
5. Ethan wants to be able to feedback into the system when he finds an answer particularly useful or unhelpful.
6. Nanjala wants to be able to show the service to her friends without explaining how to use it every time.
7. Nanjala wants to get recommendations on content she might find interesting.

Haven't
decided yet if
this is a feature
I will
implement -
time based!

4.2.2 User Requirements

User goals are distilled into formal user requirements. Each user requirement corresponds directly to the correspondingly numbered user goal.

1. Users must be able to request general or descriptive information on a geographical entity.
2. Users must be able to ask questions requesting a fact or statistic about an entity and get the answer or number they are looking for.
3. Users must be able to interact with the system over SMS.

4. Users must be able to make their queries and get responses quickly.
5. Users must be able to feedback to the system when an answer is of noticeably low or high quality so the system can learn and improve.
6. Users must be able to learn how to use the system themselves and this learning process should be quick.
7. Users must be able to get recommendations on entities they might be interested in, based on previous query history.

Haven't
decided yet if
this is a feature
I will
implement -
time based!

4.2.3 System Requirements - Functional

1. The system will receive queries via SMS, process them and then respond via SMS.
2. The system shall use a datasource that contains both facts and descriptive text on geographical entities to answer queries.
3. The system must be able to parse queries to identify the entity that is been asked about and the specific parameter of that entity that should be returned as the answer to the user's query.
4. As well as giving direct facts and statistics, the system should be able to give general descriptive information on an entity.
5. The system must be able to match words of similar meaning to the parameters in the data source. For example, the word 'long' must be matched to the parameter 'length'.
6. The system must learn from user feedback when answers that are returned are of high or low quality an apply this knowledge to future queries.
7. The system must be self-explanatory, with the facility to offer instructions to a user.
8. The system must use user state so that it can accept queries that are dependent on past questions (for example, a query asking for more information or feedback on the quality of a past question).
9. The system must be easy to implement and flexible, to accommodate future projects that make use of adaptations of this project.

4.2.4 System Requirements - Non-Functional

Fit criterion are stated after each non-functional requirement, which will be used to identify whether a requirement has been met successfully.

1. The system behind the SMS gateway must respond in a timely manner. Fit Criterion: from an SMS been received, a response must be sent within 10 seconds. SMS transmission times on the cellular network and gateway are out of the control of this project, so are not accounted for.
2. The answers returned must be of high quality. Fit Criterion: in testing, the system returns the requested information for at least 80% of queries.
3. Answers that are consistently rated poorly are no returned in fewer cases. Fit Criterion: to determine later
4. Instructions to use the service should be clear and concise, with no ambiguity. Fit Criterion: to determine later - may not keep this
5. The system should accept and be able to process questions in natural human language as opposed to a delimited message.to determine later

4.3 Evaluating Success

5 Design

5.1 Software Design

5.1.1 Micro-system for Protecting User Identity

In section 3.2.2, the need to protect the privacy of a user and the general technique that will be used (involving irreversible hashing) are described. The design of this system will now be discussed.

The main gateway into the system will be an HTTP call to a Python function, which takes the phone number of the user and the question being asked as its parameters. The header and basic structure of this function is:

```
def entry(userCell , question):  
    hashedCell = someHashFunction(userCell)  
    answer = findAnswer(hashedCell , question)  
    return userCell , answer
```

The userCell is required at the output of the function to determine the number to which a response will be sent. However, as the computation of an answer only requires a general constant representation of user identity, not necessarily the user's phone number, the specific value of userCell is not required within the computation of the answer. As such, a hash is passed into the answer-computing function. This hash will be used within the database to store a user's history and profile. This means that only a user's history can be identified from the database, not the actual identity of a user.

5.2 Language, Platform and Tools and Libraries

5.2.1 Language

The language Python has been selected to be the main language in project for four reasons:

1. Python is universally compatible and simple to execute. Once the server application is written, it will be simple to migrate it between machines and adjust the application's configuration or environment.
2. Python is commonly used and has significant community support in the form of libraries for all the tools that will be used in this project (for example, libraries for querying common databases and for sending text messages).
3. Python is an extremely flexible language in which it is easy to implement applications of this type without syntax issues. This fulfils the requirement (4.2.3 point 9) that the application is easy and quick to implement.

4. The author has significant expertise in the language, as does Lilian Blot (dissertation Supervisor).

A testing utility will also be created to query the application from a personal computer without using SMS. A simple OS X Cocoa application will be created that queries the HTTP interface with JSON in the expected format and displays an answer. Objective-C and Mac OS X were chosen as the language and platform for this as the author has experience with Objective-C and uses a Mac. There was no requirement for universal compatibility because the application is only for testing and development purposes.

5.2.2 Platform

The development platform of the utility will be Linux. This is for three reasons:

1. Linux manages Python and its libraries in a simple way without complex installers with tools like the Python Package Index (a library repository) [20].
2. Common libraries that will be used for creating a HTTP server, creating a REST API, making database queries and sending SMSs tend to be written primarily on the Linux platform.
3. The author has familiarity with Linux and has access to significant Linux infrastructure under a grant from Rackspace.

Lilian: should I explain what this is somewhere?

As previously stated, the interface to the application will be over the SMS platform.

5.2.3 Database

There are two tables of data that need to be stored. These are the tables for user profiles (to allow the application to work with stateful queries), and for the machine learning of question-answer quality. The logical design of these two tables/collections is:

Conceptual Data Model

There are five entities in the conceptual data model for this project. These are:

- **Users:** An entity that represents a user, purely by an identifier based on their phone number.
- **Complete User History:** A history of all questions they have asked previously to allow for future expansion of the system, perhaps to implement a recommendations system.

- **Last Question Asked by a User:** An object containing the parameters of the last query. This allows the system to handle stateful queries based upon a previous query.
- **Keyword Pairs Quality Learning:** Will contain the property requested by a user, the property returned to that user, and the number of times a matching was ranked as been either one, two, three, four or five stars.
- **Initial Intelligence:** Undecided.

These entities shown in Figure 7, which also shows the entity relationships.

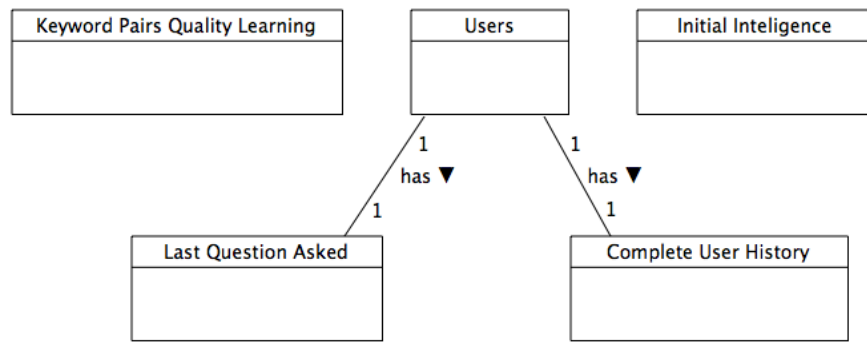


Figure 7: Conceptual Model of Data

Logical Database Design

The logical design of the database reduces the number of entities from five in the conceptual model to three. This is because there is a one-to-one relationship between the *User* entity and both the *Complete User History* and *Last Question Asked by a User* entities, with a small number of fields. In a large system where relational databases were used it would make sense to keep these entities separate, but in this case merging them allows a wider variety of databases to be used, simplifies the implementation and does not add significantly add to the complexity of the data model.

The logical database requirements are listed in Table 2 and represented graphically in Figure 8.

Database Technology

The database technology requirements for this project are:

1. The selected database must be as simple in implementation and interface as possible, whilst still supporting the system requirements. This is to allow for quick development and to avoid adding unnecessary complexity.

Need to fill in
initial
intelligence

Table/Collection	Parameters
Users	(<u>id</u> , cellNumber, history, lastQuestionText, lastQuestionRequestedProperty, lastQuestionReturnedProperty)
KeywordRelatednessQuality	(<u>id</u> , givenProperty, returnedProperty, oneStarRatings, twoStarRatings, threeStarRatings, fourStarRatings, fiveStarRatings)
InitialIntelligence	Undecided.

Table 2: Logical Database Design. Underlined fields represent primary keys.

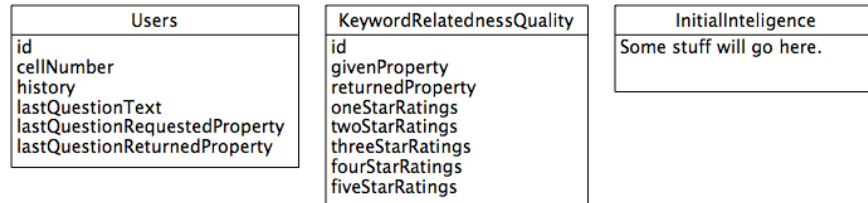


Figure 8: Logical Model of Data

2. The selected database must allow for storage and retrieval of records with multiple parameters.
3. Support for records that are related to each other is not required; so relational databases are not required, but not ruled out.
4. The database needs to have built-in support for Sharding and Replication/Mirroring to allow the application to scale.
5. The database must have a library for interfacing with it in the language of this project.

There are a number of database technologies available that would fulfil the data requirements of this project. These can be split into the categories of relational databases (SQL) and document stores (NoSQL). Two common database approaches that meet the above requirements are MySQL with an ORM ⁴, and MongoDB with the Mongo Client. Table 3 shows a comparison of both of these technologies.

From this comparison, the decision to use MongoDB has been taken because MongoDB has more flexibility than SQL, writing queries is easier in Python and, other than

⁴ORM: Object-relational mapping tool, which maps queries from the syntax of your programming language to SQL. ORM's commonly also apply optimisations to queries, and validate them against SQL injections.

	MySQL/SQLAlchemy	MongoDB
Fixed Schema	Yes, reducing flexible	No, giving flexibility
Querying Simplicity	Requires complex ORM and awareness of transactions, or construction of SQL queries.	MongoDB comes with a Python client, which allows queries in the form of dictionaries.
Relational Queries Supported	Yes (could be of help in a future extension)	No
Records support nested parameters	No	Yes
Built in Sharding support	No	Yes
Built in Mirroring support	No	Yes

Table 3: MySQL & SQLAlchemy vs MongoDB

the possibility of future extensions requiring relational structure, it is more scalable than SQL.

Physical Database Design

MongoDB separates entities into collections but does not place restrictions on the structure of objects within a collection. The general structure of the keys of a document (a JSON dictionary) is shown in Table 4.

5.2.4 Libraries

Database Querying

Creating HTTP Server and REST API

Sending SMS

Users	<pre>{ '_id ', 'cellNumber ', 'history ', 'lastQuestion' : { 'text ', 'givenProperty ', 'returnedProperty ' } }</pre>
KeywordRelatednessQuality	<pre>{ '_id ', 'givenProperty ', 'returnedProperty ', 'oneStarRatings ', 'twoStarRatings ', 'threeStarRatings ', 'fourStarRatings ', 'fiveStarRatings ' }</pre>
InitialIntelligence	Something here Undecided.

Table 4: Typical Document in each MongoDB Collection

6 Implementation

This section will contain information on how data is stored, the structure of databases, and the format of and method through which data is sent to an SMS provider.

7 Results

This section's content...

8 Discussion

This section's content...

9 Evaluation

This section's content...

10 Extending this project

Section not complete If the author was able to expand this project, areas for expansion include:

- Addressing the issue of literacy
- Automating the expansion of the database

11 Conclusion

This section's content...

**Melby reference not right, publisher weird - ask Lilian in next session. Url
Commented in TEX here**

References

- [1] U. N. international community, “The universal declaration of human rights,” 1948. [Online]. Available: <http://www.un.org/en/documents/udhr>
- [2] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, and D. Estrin, “Diversity in smartphone usage,” in *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '10. New York, NY, USA: ACM, 2010, pp. 179–194. [Online]. Available: <http://doi.acm.org/10.1145/1814433.1814453>
- [3] I. T. U. (ITU), “Key ict indicators for developed and developing countries and the world (totals and penetration rates),” Geneva, 2013. [Online]. Available: http://www.itu.int/en/ITU-D/Statistics/Documents/statistics/2013/ITU_Key_2005-2013_ICT_data.xls
- [4] O. for National Statistics, “Internet access quarterly update, q1 2014 release,” Online report, London, Tech. Rep., 2014. [Online]. Available: <http://www.ons.gov.uk/ons/rel/rdit2/internet-access-quarterly-update/q1-2014/index.html>
- [5] K. Fox. (2011, Jul.) Africa’s mobile economic revolution. [Online]. Available: <http://www.theguardian.com/technology/2011/jul/24/mobile-phones-africa-microfinance-farming>
- [6] J. C. Aker and I. M. Mbiti, “Mobile phones and economic development in africa,” *The Journal of Economic Perspectives*, vol. 24, no. 3, pp. pp. 207–232, 2010. [Online]. Available: <http://www.jstor.org/stable/20799163>
- [7] I. T. Union, “Measuring the information society,” International Telecommunication Union, Geneva, Tech. Rep., 2013. [Online]. Available: http://www.itu.int/en/ITU-D/Statistics/Documents/publications/mis2013/MIS2013_without_Annex_4.pdf
- [8] D. Kenny, “The ethics of machine translation,” 2011.
- [9] G. Inc. (2014) Google research: Find out how our translations are created. [Online]. Available: http://translate.google.co.uk/about/intl/en_ALL/
- [10] A. K. Melby, “Why cant a computer translate more like a person,” 2006.
- [11] S. Goldwasser and M. Bellare. University Lecture Notes. Accessed: 2014-11-15. [Online]. Available: <http://cseweb.ucsd.edu/~mihir/papers/gb.pdf>
- [12] R. Anderson, *Security Engineering*, 2nd ed. Indiana, United States: Wiley Publishing, Inc., 2008.

- [13] B. Schneier. (2004) Cryptanalysis of md5 and sha: Time for a new standard. Accessed: 2014-11-16. [Online]. Available: https://www.schneier.com/essays/archives/2004/08/cryptanalysis_of_md5.html
- [14] C. Teat and S. Peltsverger, “The security of cryptographic hashes,” in *Proceedings of the 49th Annual Southeast Regional Conference*, ser. ACM-SE '11. New York, NY, USA: ACM, 2011, pp. 103–108. [Online]. Available: <http://doi.acm.org/10.1145/2016039.2016072>
- [15] I. Sommerville, *Software Engineering*. Pearson Education, 2010.
- [16] N. Rahmany. (2012) The differences between life cycle models - advantages and disadvantages. Accessed: 2015-01-27. [Online]. Available: <https://narbit.wordpress.com/2012/06/10/the-differences-between-life-cycle-models-advantages-and-disadvantages/>
- [17] C. Larman, *Agile and iterative development: a manager's guide*. Addison-Wesley Professional, 2004.
- [18] B. Boehm, “A spiral model of software development and enhancement,” *Computer*, vol. 21, no. 5, pp. 61–72, May 1988.
- [19] B. Boehm and W. J. Hansen, “Spiral development: Experience, principles, and refinements,” DTIC Document, Tech. Rep., 2000.
- [20] P. community. (2015) Pypi - the python package index. Accessed: 2015-02-04. [Online]. Available: <https://pypi.python.org/pypi>

12 Appendix - Personas

12.1 Nanjala

12.1.1 Demographics

- Female, aged 15
- Living in the village of Amuria, Uganda, 300km from the Capital, Kampala. Internet access is difficult and expensive to obtain in Amuria, though there is good mobile network coverage.
- Currently in general education at her village school.
- Nanjala is curious and wants to be able to ask questions and retrieve and read more information at her own leisure. She prefers to be told about something than to learn facts.

12.1.2 Average Day

In a typical day, Nanjala will rise early to help her family with housework chores, before she prepares for school. Nanjala wants to become a teacher. At school, she studies Science, Maths and Geography, and works hard for her classes and on her homework. Nanjala's school has a small library and a body of students, all of whom are as hard-working as Nanjala, meaning that books are often unavailable when Nanjala would like to look something up. This increases the amount of time that her homework takes, and can even result in her not having access to information when she must go home to her family.

12.2 Cedric

12.2.1 Demographics

- Male, aged 27
- Lives in Zurich, Switzerland, and goes on regular hiking trips, from 3-4 day trips around Europe to longer trips to the South American Andes and southern Asia.
- Works in a service role in the city and so has little disposable income to spend on guide books for the many places he visits.
- Cedric is a competent albeit not professional hiker, often going off-track and summiting 'difficult' peaks or reaching famous camping grounds. He likes to be spontaneous, only deciding which peaks he will attempt when he arrives.
- Cedric is very conscientious and wants to contribute his own knowledge to the world; especially to projects that help him.

12.2.2 Average European Trip

A typical trip will involve an intercity train from the Zurich Hauptbahnhof (main station) to the city closest to the range Cedric will be hiking. He takes the train because it tends to be cheaper than flights. Once in the area, Cedric will take local trains and buses to get into the mountain ranges. There, he will use the local hiking trail network signs and conversations with the locals to select a peak or route. Once Cedric has got moving and is reaching higher ground, he uses a combination of his view and the signage to identify other peaks that he would like to wander around.

Cedric takes a non-smart phone with him because a) the battery life on his smart phone is not long enough for his whole trip, whereas his non-smart phone lasts a week at a time, and b) his smart phone would not be useful anyway, since the internet is so intermittent. He still wishes he could get information like whether there are maintained routes on the slopes of a given peak or the height of a peak that he sees without having to phone friends back in the city.