GenesisCare

COMP3850 - Group 16

SRS/Requirements/Scoping Document V1.03 - 20/05/21

Version History 2

Definitions, Abbreviations 3

1. Project Scope/Purpose 4

2. Data understanding 5

3. Data preparation 6

4. Modelling 8

5. Evaluation 9

6. Deployment 10

7. Requirements 11

Appendix A - Simple Test Case 12

Appendix B - Multi Test Case 12

Appendix C - Angled Test Case 13

Appendix D - Rotated Test Case 13

Meeting with Client 14

Version History

Version Number	Description
1.00 - 30/3/21	Lachlan, Shadman Preliminary document submitted for client review including: - Data understanding - Data preparation - Modelling - Evaluation - Deployment - Project Scope
1.01 - 1/4/21	Lachlan Submission for D2 including changes: - Added Project scope - Added Appendix - Added Definitions tables - Added Meeting notes - Added Requirements - Expanded on all unfinalised sections
1.02 - 29/4/21	Lachlan Submission for D3 including changes: - Fixed typographic errors resulting from google docs - Added version history Jake Submission for D3 including changes: - Revised Project Scope (renamed to Project Scope/Purpose) - Revised Data Preparation - Revised Evaluation Junyang Qian Submission for D3 including changes: - Add some details for deployment model
1.03 - 20/05/21	Lachlan - Updated dates and version number to reflect newest submission

Definitions, Abbreviations

ID	Phrase	Description
1	API	Application Programming Interface
2	GenesisCare	The client of the project
3	Team 16	The team that is developing the project
4	Python	Python programming language
5	Google Tesseract	Google's open source OCR
6	RGB and Greyscale	Colour (Red Green Blue), and Black and White
7	CV2	OpenCV project
8	OCR	Optical Character Recognition
9	OpenCV	Open source computer vision library
10	MVP	Minimum Viable Product to fulfill basic requirements
11	FP	Final product including all requirements

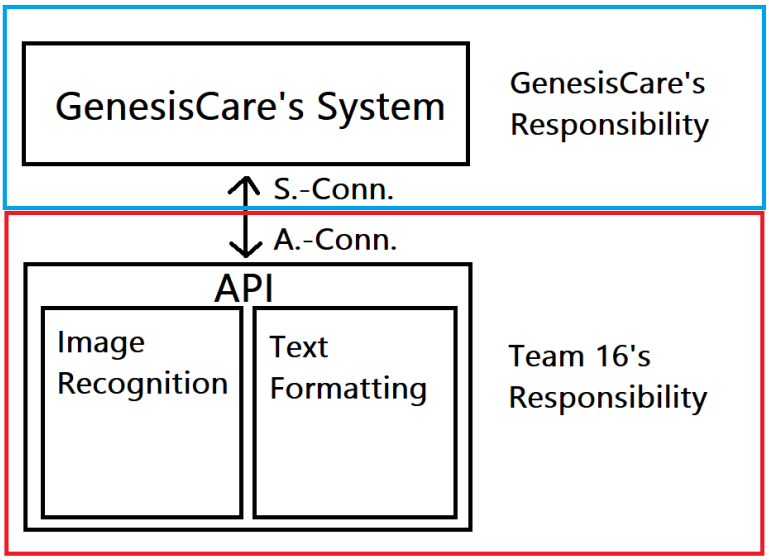
1. Project Scope/Purpose

This document is for the Medication box image recognition API that Team 16 is developing for GenesisCare. This project requires the production and integration of an API with the systems of GenesisCare, of which the former is the role of Team16, and the latter GenesisCare.

The purpose of this project is to enhance and extend upon GenesisCare's current systems, providing a further alternative method for patients of GenesisCare to upload the details of their medication. This method is efficient as it saves patients time with having to manually enter details of their pre-existing prescribed medications that GenesisCare requires before undergoing any procedures with the patient.

This proposed system will aid in greatly automating the process of obtaining crucial patient information in a timely manner for both GenesisCare and the patient.

ID	Broad Task	Description	Responsible
1	API	The API as a whole to be developed	Team 16
1a	Image Recognition	The start of the API to create text from an image of medication box	Team 16
1b	Text Formatting	The end of the API to format the text creating in 1a so that it can be used by GenesisCare	Team 16
2	API Connectivity	The API needs to be able to be accessed by ant system of GenesisCare's choosing	Team 16
3	System Connectivity	The Existing (or otherwise) System at GenesisCare needs to be modified to allow API access and integration	GenesisCare



2. Data understanding

Test data cases have been provided by GenesisCare (Appendix A)

Photos of medication boxes are going to be provided by the patients being (or going to be) cared for by GenesisCare, and genesis care will provide these photos to the API being developed by Team 16 to process them. GenesisCare has access to private medication dictionaries that are unable to be shared publicly due to privacy reasons. These have the potential to be used in conjunction with the API, but is something we are unable to test. Data will not be crowd-sourced by Team 16 as it is outside of the scope of work for this project.

The data structure is assumed (input restricted by way of GenesisCare's website) to be 1 or more photos that each contain 1 or more medication boxes in them. It is assumed this will be complete data, as in these pictures will include all relevant details of all relevant medicines the patient is currently taking. It is assumed that the photos required will not be perfect quality (as users are taking them) but that they will be of a minimum visual quality such that a human could read then, whereby we assume that our API could also then read and process the images.

This route of analysing submitted photos will be the main method by which the developed API will be able to determine the medication being taken by a patient. Use of known databases of medicines has been considered to supplement this analysis, and/or to provide an ability to auto-correct any typographic errors encountered by using a database as a word list.

The effectiveness will be verified by testing during the development phase (using the provided test cases as a dev-set for our design, with known expected results), and by having rolling monitoring of the live system, initially checking all produced results (using these as a test-set) and overtime moving to check occasionally that the API is producing accurate results.

ID	Name	Description
1	Initial data	Test set has been provided by GenesisCare (Appendix A)
2	Data Collection	Data is provided to API by GenesisCare when there are images waiting to be processed, data does not need to be collected for model training as pretrained models are used
3	Data quality	Data is photos from users so can be assumed to be of moderate quality, visually legible but imperfect
4	Data structure	Data consists of photos of medication boxes from patients (with the potential of multiple photos or multiple medications in the same photo)
5	Data completeness	Data is assumed to comprise all medications being taken by the patients, with all vital data being within the photos of the medication boxes
6	Data Evaluation	Provided user data and the generated outputs of the model will be checked manually by the development team during development, and check periodically by GenesisCare after deployment to verify the system and customers cohesively interact

3. Data preparation

The current plan is for GenesisCare to provide images of medicine boxes to the API developed Team 16. This API will label and extract from these images the critical medical information of what medicine they are and the dose the patient is one, among other listed details.

Current data preparation activities will include (GenesisCare) creating the ability for users to submit photos to the GenesisCare website (restricted to photos only), and then will include extensive modulation of these images (as the quality is likely to be poor) to extract the required information.

As the task being undertaken is not to find trends among a big-data set (pre-trained models are planned to be used), the API does not need to perform data-set wide operations such as removing outliers or anonymising data. However, considering the photos being processed are medical records, security should be the first priority of the project. Any images sent, modified, returned, or otherwise and their corresponding image-to-text analysis must either be not saved or saved in a secure location; both referring to which server is used, and where on the server the operations take place.

Current plans for data processing are based around Python and utilising pre-trained packages (such as Google Tesseract) for document scanning, image de-noising and upscaling, and text recognition; though this is anticipated to grow as a project solution plan is continued to be developed and finalised. The end-end pipeline for dataflow is:

ID	Name	Description
1	Patient	Patient takes photos of their medication
2	GenesisCare Website	Photos are submitted to GenesisCare's Website
3	GenesisCare Servers	Photos are transferred to internal servers to request processing from API
4	Team 16 API	Photos are given to API, multiple steps are taken
4a	Data pre-processing	Photos are cleaned up, through the use of technologies such as document scanning or image rotation (etc)
4b	Data processing	Photos are passed through an image to text detector to read the contents of the image into a text representation
4c	Data analysis/formatting	Text representation of photos is Processing (potentially with the help of medication dictionaries) to create a formatted file of the medications benign taken by a patient
5	GenesisCare Servers	Formatted text file is returned to the GenesisCare server for required usage (further processing, storage, details outside scope of Team 16)

The primary input to the system is images of medication, however some of these images may include multiple boxes in the same image. This will need to be handled in the Data

pre-processing stage, and will likely require a single picture to be separated into multiple pictures, before being sent through the standard Data processing pipeline. As such there is only 1 model, however it will include sub-models to process certain edge cases (E.g. rotated/angled images) as best as can be reasonably achieved.

These steps should form part of “Data Normalisation”, that is making the data available more similar to that of what the pre-trained packages were trained on, making the images more easily usable. Most pictures of text in data-science training sets are the correct orientation and have simple plain backgrounds, so because of this it is advantageous to make the data similar to this as that will allow these pretrained models to be more effective on the patient’s images. For example, in preliminary tests, off angle text and high-frequency/noise backgrounds led to large artifacting compared to the simpler and cleaner images.

The data that is processed and outcomes that are produced are not saved in the API that Team 16 is developing. Instead, the data produced by the API will be sent back to the GenesisCare Server that requested the data. As this data is Medical records, this needs to be treated in a secure manner, even though it will not be saved by the API.

ID	Name	Description
1	Data Preparation	As pre-trained models are being used, the scope of the project does not involve large datasets which would need to be prepared with normalisation, instead individual images need to be cleaned for increased performance
2	Data Processing	As pre-trained models are being used, the scope of the project does not involve large datasets which would need to be processed in batches, instead individual photos need to be cleaned and converted into formatted text
3	Data Pipelines	As pre-trained models are being used, there is only a single processing pipeline, however pre-processing sub-pipelines may be needed (pre-processing some harder photos more than others) if some test cases are too difficult for a fully unified approach, or if segmentation results in increased performance
4	Data Storage	Currently, medication photos are saved to a folder that the API then reads the images from. These photos are either jpg or png file formats. After a medication photo has been processed, the API currently saves formatted text files as plaintext .csv files. However, this may change in the future due to implementation preferences on GenesisCare’s side.

4. Modelling

The design being pursued relies on pre-trained models (E.g. Google Tesseract) so Team 16 does not need to choose a Machine Learning Model to implement, instead Team 16 needs to decide on the design of the system, which pre-processing and post-processing algorithms will be performed to increase the effectiveness of the API. The reason these pre-trained models were considered is due to the scope of the project, there is neither funds nor development time available to acquire the test-sets and model these subcomponents.

Currently this involves the use of RGB and Greyscale conversions, inverted colours, as well as simple image upscaling. These have been tested to determine what settings can work best for the given test images with the tested pre-trained model. The eventual plan is to include other features including but not limited to, document scanning and image rotation to detect off angle images, and AI-based image upscaling, and possibly a medical dictionary to provide non-standard English resources to the model

The model will be evaluated by using test-cases with known outputs before deployment, as well as periodically reviewing the results of the deployed system and manually reviewing its output

Test-cases for the provided photos need to be created with expected outputs, so that the photos can be used on the model to test its effectiveness. The tested outputs of the deployed system need to have test-cases and expected outputs created for them to compare against their produced output.

ID	Name	Description
1	Model	Pre-trained models for image detection, document detection, and AI upscaling
2	Model validity	Model will be verified by the use of test cases to confirm correct outputs
3	Test cases	Test images have been provided and must be turned into test cases to be used for testing as well as development

5. Evaluation

The results will be reviewed and validated through quality assurance checks and pre-embedded controls. Radical change management and tracking governance procedures will be proposed for tenderisation. Ensuring data is evaluated in accordance with accuracy, relevance, completeness, and consistency expectations. Enabling fast transformations and changes with associated enterprise applications. Further the digital solution will be equipped with existing enterprise tools and dashboards to detect abnormalities, outliers, broken trends, and any other unusual scenarios that may occur during data consolidation.

The current model is built around the Google Tesseract pre-trained model. If the model built around it cannot be considered business viable after considerable testing, then alternatives such as pre-trained CV2 models will need to be investigated, implemented, and compared with the existing model.

Ground truth data is obtained via photos uploaded by end-users (patients of GenesisCare) that are stored in a folder (currently local during development).

ID	Name	Description
1	Development Evaluation	Evaluation of the models during development and testing to determine business viability should be continued
2	Deployment Evaluation	Evaluation of the models after deployment to continue to evaluate business viability over a long period of time
3	Changes to model	If a model is determined to no longer make a viable business case, then the model needs to either be reworked or replaces with an alternative model that can fit the business case

6. Deployment

The model being produced by Team 16 will be provided to GenesisCare, in addition to documentation to be produced. Due to the nature of the project, additional training of GenesisCare staff will not be necessary as user interaction with the API is minimal but documentation will still be provided in the form of user manuals.

ID	Name	Date	Description
1	Feasibility Study	12 March 2021	Business Case for the project
2	Project Plan and SRS	1 April 2021	Preliminary plan for project and solution
3	Design, Test, Prototype/MVP	29 April 2021	Updated design documentation and testing as well as Minimum Viable Product
4	User/training manual	20 May 2021	Documentation for users to explain using the developed system
5	Reflective report	3 June 2021	Reflection on things achieved and not
6	Presentation	3 June 2021	Presentation of project with demonstration
7	Product delivery	3-24 June 2021	All files and documentation related to the product

Deploy:

- Following the agile methodology, the updates for the product are deployed every week.

Report:

- The results of the deployment are organized into a report for the team maintenance phase

Maintain:

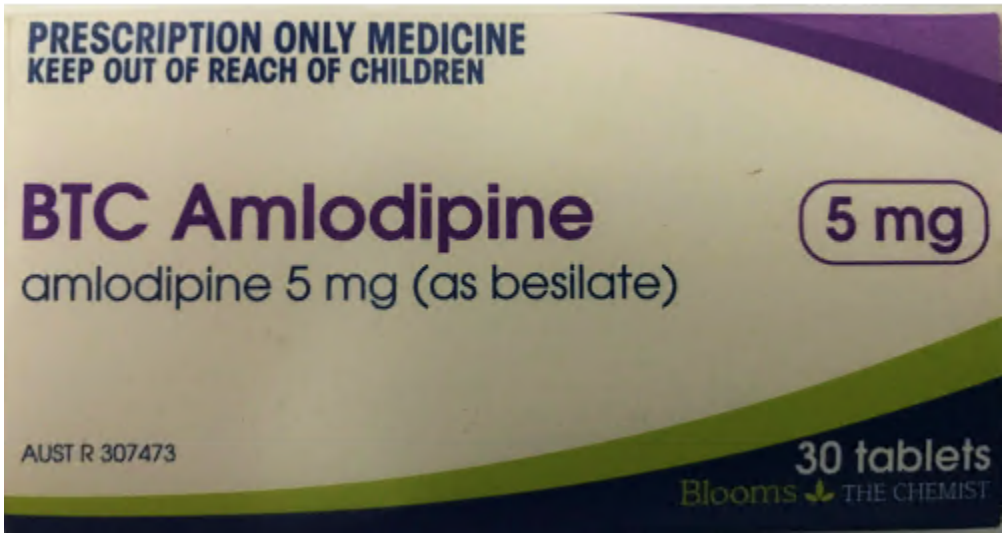
- The team will try to fix any errors or update the previous product

7. Requirements

The project can be divided into separate requirements with different completion levels. The first is the MVP which serves as a proof of concept that the system can work as intended, by implementing the important base functions of the API (limited image recognition and text formatting). The FP then includes all facets of the project including both functionality and useability by GenesisCare.

ID	Requirement	Stage
1	API must be able to read the text from a simple set of test cases	MVP
2	API must be able to read the text from a hard set of test cases including rotated, off angle, and multi-box photos	FP
3	API must be able to format some key text E.G. Brand, Dose	MVP
4	API must be able to format the read text from requirement 1 and 2	FP
5	API must be able to communicate with GenesisCare's Servers	FP
6	API must be fully self-contained (using docker or otherwise)	FP
7	API must Process a single photo in < 5 seconds	FP
8	API must keep medical files secure during processing and communication with the server	FP
9	User manuals and Installation manuals must be produced	FP

Appendix A - Simple Test Case



Appendix B - Multi Test Case

Appendix C - Angled Test Case

Appendix D - Rotated Test Case

Meeting with Client

Meeting date and time:

Tuesday 30th March 2021

Feedback received:

“Group 16 Feedback

Suggest using 'API' as it is an acronym - easier to read

2. Great answer to the question, clearly demonstrated a deep understanding of the problem, potential blockers and solutions, as well as extra considerations.

3. Likewise as above, good job.

4. Correct - as is often the case with the types of products we develop, the hardest / most complex piece of work is the change management (and also safety & quality team).

5. Correct - we expect user manual & install manuals. (essentially just a README is fine). I would advocate for a Docker solution but will leave that to you. “

Team response/action points

“We’ll take that advice, and also add an appendix and definition/abbreviation section”

Additionally, Project Scope, Appendix and Requirements sections were added