3. Project plan and Requirements Specification

This section outlines the initial project planning stages as well as justification for requirements gathering methodology and initial specification. As the product is a work in progress, some of the initial requirements may change through the product lifecycle.

3.1 Stakeholder Identification

This project, just like all others has 2 kinds of stakeholders: direct and indirect. The direct stakeholders are those closest to the project where the indirect are possible end users. The proposed project however, is research focused and as such the main goal of the project is to further the development of 2D recognition technology using recent advances in the field of machine learning. It is because of this, that the main stakeholder listed is the author of this report. All other stakeholders are indirect. These are listed in the table below.

Stakeholder ID	Name	Role
D01	Adam Grimley	Developer / Author
I01	Dr Gaye Lightbody	Mentor / Advisor
I02	Kainos Innovation Team	Advisors
I03	PSG	Peer support / Focus group

Figure 3 - Stakeholders

If required by the author, discussions with specialist machine learning developers as listed in section 1.1 of this report may be consulted. The author does not intend for this to be required however if they are consulted, they will be added as an indirect stakeholder in a consultant role.

3.2 Requirements gathering methodology

In gathering requirements, a meeting was placed with members of the Kainos Applied Innovation team to discuss what would be required of a system like that which has been proposed. The team are recognised members of the NI research and development community and as such, have experience in projects such as this. The author conducted this meeting using carefully selected questions designed to create a discussion around the proposed application and to identify and examine any unforeseen edge cases that may come from standard use of the product.

Going forward in the development of the application and research, the author has gathered a focus group to further develop requirements and to receive feedback about the product. This has shown to be a non-discriminatory way of increasing productive input during the development of a product [10] and has been chosen by the author for this reason. These meetings will give people a safe environment to raise concerns about the product direction and to suggest ways in which the product can be improved.

From these focus group meetings, the suggested changes will be taken on board, checked to see if they can fit into the project scope, and passed through as a change request. These requests are examined by the author and checked against the schedule of development to see if they are valuable and time efficient. These may also be discussed with the author’s mentor to examine whether the changes will add tangible increase in the project viability. Each requirement listed above has been taken from the initial meetings with Kainos and the focus group and may change or be added to as the project is in development.

3.3 Requirements prioritisation strategy used

After being introduced in university, the author decided the best way to prioritise requirements would be to use Karl Wiegers Relative Weighting, evaluating each requirement based on cost, value and risk [11]. In the case of the proposed project, the cost value will represent the estimated time for each requirement to be fulfilled. The main component of the prioritisation strategy is the equation:

Value percentage / (cost percentage * cost weight) + (risk percentage * risk weight)

This equation allows us to prioritise each requirement based on weights of how important the cost or the risk is. Due to the high risk of this project, the risk weight is set slightly higher than the cost as the author believes time to be less of a factor than the technical complexity. An example of the relative weighting used is in figure 4 below.

Relative Weights:	**1.0**	**1.0**			**1.0**		**1.2**		
	Relative Importance	**Relative Penalty**	**Total Value**	**Value %**	**Relative Cost**	**Cost %**	**Relative Risk**	**Risk %**	**Priority**
Totals	17	22	39	100	19	100.000	13	100.000	
User does not process the image locally	4	6	10	25.6	5	26.316	3	23.077	0.475
System will hold standard 3D models of common objects	5	7	12	30.8	8	42.105	2	15.385	0.508
System will recognise and report the relative rotation of desired object	8	9	17	43.6	6	31.579	8	61.538	0.413

Figure 4 - Example of Relative Weights

This process shows which requirement should be handled first. In this case, it would be the 3D models. Although they have a low risk, the time required to generate both the models and the data structure required is very high according to the author.

3.4 Initial Requirements

Requirements listed in the table below are split into FR (functional requirement) and NFR (non-functional requirement) IDs and linked to the objectives listed in section 1.3 of this report.

ID	FR/NFR	Requirement
01	FR	Application is standalone and does not require a browser
02	FR	Application takes single photograph as input
03	FR	User does not process the image locally
04	FR	Develop a public endpoint server that allows connection to ML models from anywhere

05	FR	System shall be trained using common objects
06	FR	System will hold standard 3D models of common objects
07	FR	System will recognise and report the relative rotation of desired object
08	FR	System will recognise and report the depth of desired object relative to viewpoint
09	FR	The system will automatically communicate between different components of the pipeline
10	FR	The system will substitute 3D models in virtual space based on where the specified objects are in the input image
11	FR	The user should be able to show/hide metrics during execution
12	FR	The user should not have to leave the application during the process
13	FR	The application should have Virtual Reality support
14	FR	The application should allow the selection of objects to create in the 3D space
15	NFR	System will be developed using cutting edge open source frameworks for ML
16	NFR	System will have a way to report metrics including stage times, accuracy percentiles, etc.
17	NFR	System will not hold any data that isn't marked as training data
18	NFR	System shall have a minimum of 70% test coverage
19	NFR	The API that receives and returns data should be secure
20	NFR	The application should be desktop only

3.5 Proposed System Architecture

As an example of the architecture of the system going forward in planning, the author has outlined below how the system will be structured.

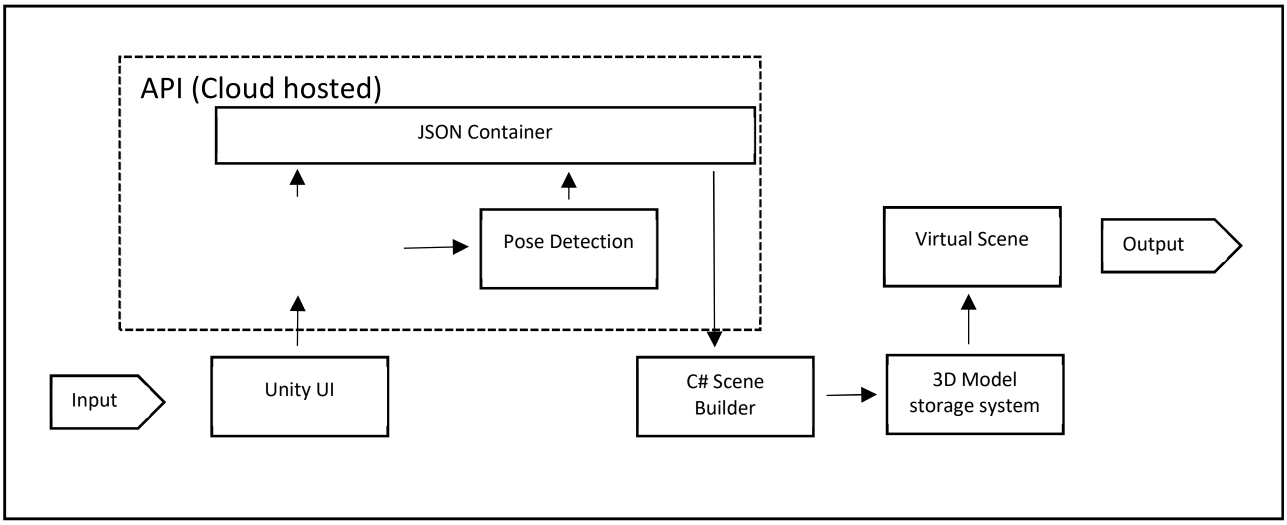

Figure 5 - Proposed Architecture

3.6 Software Lifecycle Methodology

As outlined above and throughout this report, the full system proposed is comprised of many different components. Each component should in theory be able to work independent from the others however it is the connections between each component that make the system novel. Even so, because of the independent nature of each component, the author has decided that rapid prototyping will be the best practice going forward, so that each component can be prototyped and connected from the ground up in similar timeframes.

It is because of this decision, that the methodology chosen for the project is spiral development. This methodology was first introduced by Boehm in 1988 combining waterfall and RP development. The justification of this is the timeframes of the project set by the university. These given timeframes lend themselves more naturally to a waterfall driven development life cycle which would not be as efficient for a single person team to implement into this project.

The spiral methodology gives more forgiveness to complex systems [12] such as that proposed and allows for faster reaction to unforeseen technical complexity. This is required in the project as some of the technologies that will be used are still experimental and not fully tested. The methodology proceeds as cycles of development, at the end of which there will be a product prototype. At the beginning of the cycle the author will evaluate risks associated and react accordingly – developing the highest risk components first.

In association with the spiral methodology, a Kanban board will also be used. This will be held online and hosted by GitKracken. The Kanban board will be used for tracking the progress of each development round. The proposed rounds are listed in figure 6 below. As the project is still in the planning stages, these rounds may vary slightly in the later stages of development.

Cycle	Task ID	Effort	Description
1			**Preparation**
	01	L	Proposal
	02	H	Initial project plan and report
	03	M	Research into component technologies
	04	M	Requirements development meetings
2			**Basic API**
	05	L	Setup cloud VM with public access
	06	H	Train RCNN model with single object
	07	H	Train Pose model with single object
	08	M	Move models into the cloud
	09	L	Connect models using JSON container
3			**Basic Unity integration**
	10	L	Build initial UI
	11	L	Integrate the posting of inputs to the VM public IP
	12	M	Build initial Scene builder to parse returned JSON
4			Model Building

	13	M	Build initial simple object recognised by API
	14	M	Generate structure to hold models
	15	M	Integrate structures with scene builder component to spawn objects
5			**Additive Development**
	16+	M	Train API with additional model
	17+	M	Create appropriate 3D model
	18+	M	Integrate model with scene builder component

Figure 6 - Cycles outline

In figure 6 above, cycle 5 will continue to be repeated as in theory all systems will be connected by this cycle and should just need added to. The author only expects one of these cycles to be completed however there is space for continued addition if the time is available.

Each cycle will begin with a meeting with stakeholders to outline and examine any risks associated and will end with all systems tested to assigned standards.