Final Report

Group H
Cleon Grant
Ethan Newell
Tanzum Begum
Jibril Omar Ali
Ifrah Shahid
Liban Farah
Shah Ali
Project supervisor: Frederic Fol Leymarie
Goldsmiths College, University of London
Course code: IS52018C
Software Projects
30/04/2018

Table of Contents

1. Introduction

1.1 Motivation

InteriAR is an accessible interior design tool. Our application allows users to draft their dream room; combining augmented reality techniques with carefully selected decorators who can make their augmentation a reality.

The idea initially spawned from the concept of connecting users with local handymen in an Uber-like on demand service with fast response times for any small tasks. However upon exploring further, several major potential issues arose. The primary ones were: concerns over regulating trusted handymen; already established competition in the form of “Trust a Trader”, and lack of scope for scalability and complexity.

This evolved into an application which would enable users to select 3D models of furniture and drag them into a live view of their living space, complementing their existing layout and exploring new designs. However we wanted to differentiate ourselves from any similar applications on the market; so we persisted with a match-making service and decided to partner up with local decorating companies. These decorators would be able to advise our users and implement any future augmentation options such as wall colouring, lighting installation and more.

We decided that focusing the app around augmented reality alleviated any concerns over lack of complexity, as it’s an emerging technology with a great deal of development currently within the industry. It also allowed us to consider and plan for scalable implementation within new hardware projects in the future such as Microsoft’s HoloLens.

1.2 Overview

In the following report, we will be covering a detailed description of the development and evaluation of the InteriAR application. It will begin by outlining the technology stack and the reasons for selecting each fundamental component. We will also reflect on what we believe went well and critique the less successful side, while focusing on what we have learned for any future projects going forwards.

Following this will be the formative evaluation, detailing our user testing and describing how we progressed from frequent informal testing with early versions into more extensive and formalised sessions. We will point to the useful information and suggestions we took away from each stage of the testing process and summarise how it helped to form our app in its current state.

We will then walk through the design of our application, including discussing what we settled on for our minimum viable product and where we are at in the grand scheme of our overall idea. This then leads into how we’ve ran in-house testing to ensure what we have so far works to a satisfactory level.

Finally, we will conclude with a summative evaluation of our final software and also outline several ideas we have for the future development and scalability of InteriAR.

2. Development Record

2.1 Technology stack

The first step in the development process was to select an appropriate technology stack for the task. What we required was: a tool capable of handling 3D objects and preferably building to both Android and iOS; an augmented reality library compatible with this software; a database management system to host on the cloud, allowing us to insert and edit documents via scripts; and several minor utilities for extra functionality within the application such as helper libraries.

We set about deciding upon these tools primarily based upon our functional prototyping, outlined in the proposal, alongside the research conducted at the time (1).

The core development for interiAR was done in Unity3D (2), as it provides an intuitive and powerful environment for manipulating augmented 3D objects. Despite this being a brand new tool to all members of the group, we were able to pick it up relatively quickly owing to the similarities between Unity's native C# and the more familiar to us, Java (3). Once we became accustomed to the GUI heavy approach of Unity development, we were able to work and test efficiently, making use of helpful tools such as Unity Remote for instant visualisation and debugging of new features.

To complement Unity, we decided to use Wikitude (4) for the augmented reality library. We came to this choice through previously using Vuforia (5) in the functional prototyping stage and deciding that our project would benefit from a tracking based AR rather than recognition. This change allowed us to remove the need for physical markers while simultaneously enabling our drag and drop furniture design.

Another considerable task in the deployment of this application was building the back-end database. For this we followed our functional prototype and implemented a mongoDB system hosted by mLab (6). We decided to go with mongoDB primarily due to our familiarity with Javascript Object Notation and strengthened by further reading into the comparisons between it and the alternative MySQL (7).

While we are pleased with how the final database build turned out, the development was far more difficult than anticipated due to not being able to interact with the database using python as we are familiar with. This was an unforeseen downside to being constrained to C#, exacerbated by the outdated and limited documentation for Unity to mongoDB interactions.

We were further assured of our choices when speaking to representatives from Sky at the employer's fair hosted at Goldsmiths. They had recently used a very similar technology stack for a small augmented reality project of their own; encouraging our decisions and giving us confidence going forwards.

2.2 Development

Once we had our software tools selected, the next step in preparation for building our application was to agree upon and commit to a development methodology for the process. As previously planned in the proposal and bolstered by further learning on the subject, we persevered with an agile (8) manner.

In order to begin the process in an organised manner, we started off by creating an overall product backlog on Trello (9), splitting up and outlining the elements we had planned for the app (Appendix A.1). In addition to this, we created and updated the milestones in our Gantt chart (Appendix A.2), although these were modified throughout the term due to circumstances often outside of our control. We then began to create and work off user stories which helped focus the team on the higher priority aspects of the development (Appendix A.3).

We thought it was important to stick closely to the core agile methodology and selected two key principles to focus on maintaining throughout the process. "*Deliver working software frequently, from a couple of weeks to a couple of months, with a preference on the shorter timescale*" (10). Due to our limited time, we opted for the shorter end of this spectrum and implemented week long sprints for which we created individual sprint backlogs (Appendix A.4). This meant that we were held accountable to deadlines for real working progress on the software. These short sprints also helped us regularly reassess our methods and the feasibility of implementing all the features we had planned. A further major benefit to consistently producing usable software, was it allowed us to regularly hold informal user-testing sessions throughout early development. These then transitioned into larger more formal sittings; both of which will be discussed at length in the formative evaluation.

Perhaps the most vital guideline we imposed was "*The most efficient and effective method of conveying information to and within a development team is face-to-face conversation*" (10). We achieved this by compromising between an ideal of daily meetings, and the fact that we hold other responsibilities as students, and settled on consistent scrum meetings three times a week (Appendix A.5). This ensured all group-members would work hard throughout the week to progress through their individual tasks and gave an opportunity to share updates and raise any concerns or issues that had arisen.

Reflecting back on the development process; we consistently operated smoothly and efficiently. In particular, we organised our team by dividing up ourselves based upon the differing skill-sets, allowing consistent progression where no team-member was stuck doing nothing. In large, this was due to our strong commitment to maintaining the regular scrum meetings and being attentive to the weekly sprints.

A steep learning curve was the ability to combine separate libraries, unfamiliar software and our own code. The fact that we managed to combine several libraries and assets, proves that learned and will benefit from this experience. This required us to heavily utilise any provided documentation, which in some cases was extensive and sometimes very poor. We also had to fully read through and digest any third party AR code we were implementing in order for us to add to it and make changes where necessary. An example of this was taking the initial Wikitude system of drag and dropping 3D models, and adding in our own code to allow us to track the most recently selected object. This then allowed us to write a custom delete function as well as integrate Lean Touch (11), an entirely separate gesture library which we used to adapt our limited time-based model rotation with intuitive finger twisting motion.

While we are pleased with how the process went, there were several issues we encountered; making us better equipped for future projects. The heavy focus on always progressing and having functional improvements did occasionally lead us down time wasting routes. There were instances where we would be working on add-on features for the application under the assumption it would be trivial to then implement it into the core app. This was not always the case and at instances lead to an entire sprints worth of work having to be restarted.

Another issue was due to our lack of experience in group development. We found it very hard to gauge how long certain features may take to implement and what was achievable in the overall time frame. This lead to us being not entirely sure of what our minimum viable product should consist of, and perhaps being overambitious in our initial proposal. We believe this is better than the alternative of being too conservative, however in the future we feel we now have a better starting point of estimating feasibility and scope.

Finally, we would utilise version control more effectively than we did in this project. Initially we had some issues working with our Unity project in git due to certain files exceeding the maximum size allowed. This led us to explore other options and we discovered Unity Collaborate, a version control and sharing service provided by the platform themselves. At the time this seemed perfect, and for the most part it did work and served all of our needs. However due to it still being in the early stages of release and our non-premium access, it let us down in several aspects including limited “seats” for group members and the lack of the ability to entirely revert commits. For our next project we would work harder to ensure we could use git properly, which in hindsight we believe would have been possible with the correct initial set up and use of gitignore.

3. Formative Evaluation

3.1 Preliminary testing

Throughout the early stages of development, the team decided to hold semi-regular informal tests with small groups of 2-4 people. The purpose of these was to uncover potential bugs and generate early ideas about minor design tweaks. This proved helpful as it encouraged us to constantly keep the users' needs in our minds whilst developing.

3.2 Formal testing

Once we were more confident with the augmented reality implementation within our app, we began formal tests process. Figure 1 shows the state of our AR camera at this stage of development.

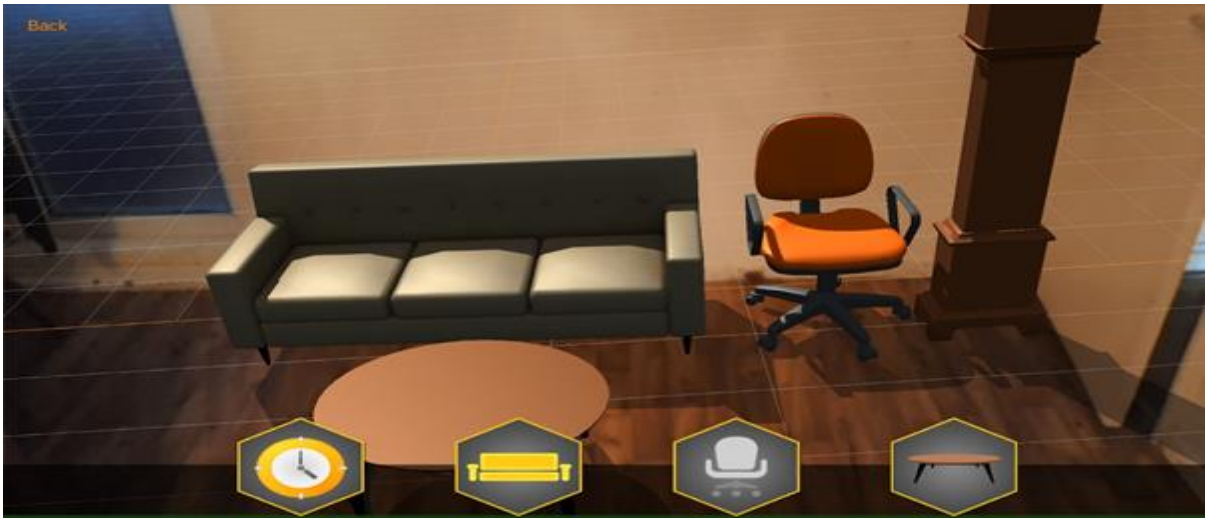


Figure 1. interiAR initial design for user testing

The main objective of the testing was to verify whether the user could determine how to intuitively use the furniture positioning features. We were also looking for feedback on elements that we could improve upon, hopefully saving us some time with further developments.

3.3 Overall objectives

We interviewed and gathered data from the users performing tasks on the app. Our intentions with these observations were the following:

- Ascertain a better understanding of how a new user behaves without any form of aid or instructions.
- Identify potential improvements in terms of the layout and design of the camera screen.
- Gather feedback on features that might be useful to implement in the future.

3.4 The task

User brief

The users were briefed about the premise of the app but weren't given a tutorial on how to use the features. During this time the user was asked to 'think aloud', meaning that they talk in real time about: what they think is happening; what they are intending to do; as well as why they made a certain action. When analysing this data, we took into account the possibility of our multitasking test requirements slightly compromising the concentration levels of the users.

End goal

Each users' end goal is to display all the furniture that is available to them in the layout of their choice, however one of the furniture items must be duplicated (e.g. 1 sofa,1 table,1 clock and 2 chairs). This was chosen in order to increase the likelihood of the participants using and testing the 'back' button.

Method of gathering data

In addition to thinking aloud, we encouraged the users to ask questions while they were performing the task. This enabled the observer to note down key thoughts of users that are new to the system and derive qualitative data based on their behaviour. We also asked the users to fill out a survey at the end of their sessions which provided the quantitative data also.

Location

Due to privacy concerns with our app involving the camera feature, we decided against performing field testing in users' homes. In lieu of this, we provided a controlled location where the observer could govern the conditions, such as the method and task time. The premise remained the same as we had maintained a consistent environment in a room where the users could concentrate without distractions.
The participant would be using the Samsung Galaxy S8 provided by us running Android Oreo.

Consent

The users were issued with a consent form that they signed before participating in the test. This stated that they were volunteers and they may leave at any time during the session (Appendix B.1).

3.5 Evaluating results

The team acknowledges the constraints of our method of testing. One of those being that the testers wouldn't be able to replicate the familiarity to the room that a genuine user would feel. Having to get familiar with a new app as well as a new environment may alter behaviour slightly. However, despite this we believe that the data that we gained from this session was very useful in understanding where we were in the development process. Not only that but the feedback and suggestions helped focus our efforts going forwards.

Explaining the premise of our app, we asked participants to explore the AR features and surveyed them on their opinions (Appendix B.2). Although the placement controls seemed simple at the time, we still expected at least a few users to take a while to get used to the gestures. Despite our

reservations, for the most part, the group found it relatively easy to use the drag and drop feature as well as scale and position them into places of interest.

The heuristic approach of the testing was something that we found to be very beneficial as it allowed us to detect what the users expect to see when downloading an app like ours. Some users even instinctively tried a twisting motion to rotate the furniture which we found to be interesting and is something that was not yet available. The users overall seemed very excited by the app, albeit partly due to it often being their first times using AR.

Upon asking for improvements, 3 out of 13 users mentioned the ability to rotate the objects to be a necessity. Fortunately, this is something that we intended to implement soon after which is a good indication that we understood what our users needed.

Also, when asked about possible UI changes several users stated that the text in general was "boring", unappealing or difficult to see. We were inclined to agree with this and as a result, we decided to implement icons instead of text buttons.

As well as this, a portion of users mentioned they had difficulty in deleting furniture they no longer wanted. Some users either didn't mention or notice the 'back' button until the team brought it to their attention. This demonstrated that we had to make it clearer to the user about the option to reset the furniture. We originally thought that we could possibly make the button read 'reset' instead of 'back', however after deliberation we chose to implement a simple 'delete' icon with an image of a bin. As per the user suggestions, this included an option to remove only the last selected object, rather than reset the whole design.

Figure 2. interiAR user interface improvements after user testing

After taking user feedback into consideration, our improved user interface is shown above in Figure 2.

After completing our AR user testing, we turned our focus to the login system and profile page. Once these passed in-house tests we hosted a similar user testing session. We asked participants to complete a series of tasks and recorded the results which can be found in the appendix (Appendix B.3).

4. Design and Implementation

We will now go through the core components of our app step-by-step and provide an overview of the design and implementation of each. This will include portions initially put forward in the proposal which didn't make it into the minimal viable product for various reasons.

The initial version of interiAR will include the following:

4.1 Login and registration

Design

From our conceptual prototype to our minimal viable product, the design has changed slightly to accommodate the change in implementation. We initially intended to use Google and Facebook OAuth (12) to handle user login, relieving us of the responsibilities that come with storing account data. However we discovered not only was it frustrating to integrate with Unity, it is also a paid service so we decided to design our own account creation instead. This of course required us to change our UI as we had to make a 'registration' form (Appendix C.1) requesting users to enter in the required information as well as a slightly edited login page (Appendix C.1).

It also required us to reconsider how we were going to maintain the data's integrity as well as keep it secure.

The overall design of this page will be reworked in future to look more pleasing. For our initial app we focused on usability and functionality rather than aesthetics.

Implementation

We chose mLab to host our database and defined a user collection containing all relevant information. Once we had implemented the UI elements of this to go with the database, this brought on the second version of our app. This was a significant change as it meant we now had the ability to store data over separate login sessions. As explained in our project proposal, we would have preferred to avoid handling sensitive data to minimise any potential data breach. To combat this we have been very careful to ensure all user passwords in our databases are hashed (Appendix C.1), protecting the sensitive data of our user base.

4.2 Snapshots and profiles

Design

Our design for the profile screen has remained very similar to what was initially planned. This is because it has not been affected by any of the other changes we have made to the system. Because of this, it follows the same style as shown in the conceptual prototype. The current design now is the user's information is displayed above and a gallery of any previously taken snapshots shown below.

These snapshots can be enlarged and displayed in an album view allowing the user to flick through any stored images (Appendix C.2).

For later versions, we plan to have more of a recurring colour scheme and theme throughout the app to make it seamless and presentable. We also plan to present the data more efficiently as we feel some of this information is unnecessarily displayed on the users landing page.

Implementation

The information displayed in the profile is drawn straight from the associated object in the cloud-hosted database (Appendix C.2).

We have added a button onto our user interface allowing the user to take snapshots of the augmented reality view. This is the picture that would eventually be shared with the decorators upon being matched. This is saved onto your profile which you can view when you log in from any device. In future we would like to implement the opt-in ability to share these on a public profile should the user be so inclined.

This snapshot functionality ended up being more complicated to implement than expected. This was due to the asynchronous nature of the "CaptureScreenshot" function meaning we couldn't simply hide the UI, take a snapshot and then re-enable it afterwards. In order to achieve the desired result of a snapshot without the interface, we had to implement co-routines calling IEnumerators which was an entirely new concept to us (Appendix C.2).

4.3 AR camera

Design

We altered the design of the camera screen over several versions. One design decision we made was changing the clickable areas/buttons from text to intuitive royalty free icons (Appendix C.3). This was brought about by our user testing and was a well-received change.

We have kept the design of a collapsible side menu holding different categories of selectable furniture in our current version as we believe it's the easiest way to navigate through lots of furniture within a restricted screen size.

This design will likely be adapted if we continue developing the app as it can be improved to being more aesthetically pleasing as well intuitive in terms of finger gestures. In terms of design we consider this to be the most vital scene as it's the primary focal point of our app.

Implementation

We have stuck with Wikitude as our Unity library to facilitate the augmented reality aspects. We have also included a unity asset package called Lean Touch which aids with the rotation of our 3D objects, linking it to finger twisting gestures. The only minor change we have made is how we handle rotation, currently, due to implementing extra features; we have 2 fingers for scaling and 3 fingers for rotation. This is not ideal as it's become natural movement for 2 finger pinch to change the scale of the object and 2 finger twist to rotate the object, but is very frustrating to accidentally scale when trying to rotate. With our next version, we aim to improve upon this with more intelligent detection of the user intentions.

For our current release, we have chosen to only include a few select IKEA items from relevant categories. This was decided as it showcases a proof of concept and could easily be expanded upon.

A feature we were unsure of implementing but upon user-testing decided it was a necessity, was the ability to delete an individual object. The way we implemented this was to track the last object interacted with by the user. This then allowed us to create a simple delete function to remove the furniture from the virtual world (Appendix C.3).

There were also certain elements not fully developed for this initial version of the app:

4.4 Wall colouring

We have decided to leave the wall colouring out of the preliminary version of the app due to time constraints and struggling with processing efficiency. This is because it turned out to be far harder than anticipated to improve upon from the functional prototype without ruining the smoothness and real-time usability of the app. We do, however, have every intention of including it in later versions.

4.5 Chat

We ended up taking the decision to exclude the chat from the MVP due to difficulties integrating it within Unity. We prioritised the AR and database elements over this as it was always intended as a bonus feature with time permitting. We are still developing this on the side with SendBird (13) and it will be one of the first features implemented.

4.6 Payment system

The payment system has been prototyped and developed outside of the app. As we have not yet developed the matching service this not been combined with our Unity project, however it is ready to be implemented when necessary (Appendix C.4). It is a very simple HTML based link to the PayPal system.

4.7 Decorator filtering system

As mentioned above, we have excluded the filtering system from this version completely as it was a large endeavour that we unfortunately had to forgo due to higher priorities. It is still of course a vital part in our overall plan, but will require further resources being put into development.

Although we didn't complete everything we set out to achieve, we still have a very functional product that can showcase the important features that lay the foundations of interiAR. These range from securely logging in our users to allowing them to take and display pictures of their newly designed rooms.

5. Ethical audit

5.1 Data protection and privacy

We have taken great care throughout the process to ensure we adhere to the Data Protection Act (14) and the soon to be enforced GDPR (15), acknowledging our role as both the data controller and processor. We have attempted to minimise any sensitive data we hold on users where ever possible. Although we ended up storing passwords despite previously having planned against it; we only deal with hashed strings, storing no part of it in plain text.

Going forwards with future development on this project, it would be a long term plan to utilise data such as market trends on furniture selected, paint colours chosen and more. However if we did go down this route we would make sure all data was anonymised and only stored if the user strictly agreed to it as we believe it's vital to not compromise trust between your platform and user base.

5.1 Intellectual property

In terms of intellectual property, we have made sure to abide by any licensing on all software and assets utilised. For example our current selection of IKEA furniture was imported from a data set listed as free to use providing we cite their research paper (16). When it comes to software tools, we've made use of educational licenses where available and were provided licenses personally for Wikitude after telling them of our project.

The only third party code we've made use of is components of the provided Wikitude and Lean Touch libraries where we have utilised, adapted and combined sections of built-in features.

6. Quality Assurance

Throughout development we made sure to maintain the integrity and validity of our software through vigorous testing, which will be outlined in this section. All tests referenced in this section were collected via a table and can be found in Appendix D.1 while extensive results/evidence of these tests can be found in Appendix D.2.

6.1 Initial Requirements

Functional testing

- Appropriate error messages should be displayed where necessary.
- All buttons should function accordingly.
- Gallery should display all images taken by user and fill up all available slots. Screenshots that are taken subsequently should not overwrite the images within the slots.

Non-functional testing

- Our database must only contain hashed versions of passwords.
- All screens within the application should be user-friendly.
- Our database should be able to store over 100,000 user accounts.
- AR camera should be compatible with majority android devices.
- Our app should be able to handle multiple users at once.

Static and dynamic testing

- There should be no defects or redundancy in any of the C# scripts.
- Any unit testing should produce no errors.
- Integration tests should run successfully and produce no errors.

6.2 Our approach to quality assurance

To maintain the desired level of quality in our application, we came up with various scenarios that we tested for in our functional and non-functional tests. These scenarios are primarily based on areas that we considered to be more prone to errors, and aspects that we were less confident with.

Another way we maintained the standard of work we strove for is by coming up with various possible defects that we checked for in our static and dynamic tests. The aim of this was to find bugs within our C# scripts and compare expected outputs with actual results to ensure they matched up. This helped clean up our code, remove errors and prevent more bugs from occurring.

6.3 How well our final system conforms to our initial requirements

From the results of our tests, we can see that some failed to meet our initial requirements; our functional tests suffered several failures. One such example of this was failing to display an error message within the login screen when the user inputted an invalid username and/or password. We were able to simply rectify this error upon discovery, thus highlighting the importance of software testing as it allowed us to find and fix a previously uncaught error. Another failure arose in our profile page. Screenshots taken by the user within the AR camera would overwrite prior images stored in the gallery when a user utilised this feature over multiple log-in sessions. This was an unexpected occurrence and unfortunately something we were unable to fix due to time constraints.

Thanks to the non-functional testing, several other issues were brought to our attention. One of these was assessing usability, where tested the user-friendliness of our UI. We found that our account creation screen forced users to input an overwhelming amount of text, making it frustrating to use. We fixed this issue by simply reducing the number of input fields and altering the database accordingly. A separate complication surfaced in the compatibility test, where we found that some Samsung phones such as the S7 Edge and S8 were not able to render the AR camera properly, instead displaying a blank white screen. We were able to fix this rendering issue by adjusting the multi-threading options within the Unity pre-build settings.

Through static testing, we were able to find defects within our C# scripts. Generally, it would be issues such as unused variables, redundant code and methods in need of refactoring, all of which we were able to easily fix. In some cases, we found string variables that would produce a NullPointerException with invalid user input. These strings query our Mongo database for specific reasons (e.g. to find whether the inputted username and password exists in the collections), and if it doesn't match up with anything, then it returns null. This was rectified by implementing Boolean statements to toggle error-messages.

We carried out black-box testing for the majority of the AR camera scene, often having to rely on manual test runs rather than automated unit testing. This was due to the level of interaction necessary and the importance of user experience. For example, ensuring all furniture collision boxes were accurately positioned. This was facilitated by the use of Unity Remote which allowed instant visualisation of any changes made without having to rebuild the entire app.

6.4 Changes made from initial requirements

The only change we made to the initial requirements was in the non-functional testing section, where we initially stated that there should be a backup database. We found out that we would need to pay over twenty pounds to have a backup in MLab.com (where our Mongo database is stored). Paying for resources is something we all agreed to not do, and as a result this was erased from the initial requirements; however in future should this product be launched this would be re-evaluated.

7. Summative Evaluation

7.1 Motivation and research

The primary goal of our project was to create an augmented reality application to allow homeowners to visualise design ideas in their own home.

In the early stages of conception we conducted extensive market research to identify whether our proposition was a viable idea with room for growth and a genuine market of interest. This involved speaking to potential stakeholders and collecting survey results which were presented in the project proposal.

This research returned positive results and also helped us identify the core stakeholders of the application which was crucial information throughout development. These were narrowed down to include: homeowners looking to redesign their living space; decorating companies searching for new clients who have ready-made plans of what they desire; and furniture production/retail companies hoping to sell their products in our system.

To gauge the potential growth of interiAR we carried out further research into multiple industries, ranging from the emerging augmented reality market to more traditional furniture companies. To make an augmented reality application we wanted to be confident that the market wasn't simply a novelty and would remain relevant in the long term. Our research showed that according several sources the worldwide augmented reality market will be worth more than 'USD 133.78 billion by 2021', growing at a rate of '85.2% until 2021' (17). The main furniture retailer we identified was IKEA; IKEA is one of the largest furniture producers in the world with a market share of 8.2% in the UK in 2016 (18). Due to this we have taken our current furniture library from the IKEA catalogue; allowing the user to visualise readily accessible items.

7.2 Evaluating the process

All group members were involved in the software development, testing and documentation of the project, endeavouring to abide by the Agile development methodology. Agile development allows flexibility within the project whilst ensuring a pragmatic approach is taken. For us this meant we arranged the group, often into pairs, to work on different parts of the project simultaneously. To maintain steady communication between members we held three scrum meetings weekly as part of our recurring short sprints. We also gathered feedback from stakeholders as the project progressed and implemented changes according to trends in the acquired data.

InteriAR's primary advantage over other home design applications is its innovation in utilising augmented reality. This means that the augmented reality must be reliable and run smoothly. To pursue this we favoured Unity3D as it is designed for, but not restricted to, 3D games. Unity is an excellent 3D engine that can be applied in many ways outside of game development, making it the

perfect software to use. Additionally, it provides options for Android and IOS deployment. The current state of the app is fully functional on Android but has problems with database connectivity on iOS. Now we have reached a minimum release, we believe Unity was the appropriate choice to the alternative, Android Studio. This is due to our extensive reliance upon 3D models and their lack of native support from the aforementioned option.

Throughout development, we were pleased with our choice of database management system due to the flexibility of documents in a noSQL environment. Presently, as a result of other modules, we have gained experience with MySQL thus would be comfortable using a relational database in future. The main struggle with implementing mongoDB was the lack of documentation for Unity integration. In terms of database security, on the basis of an official release, we acknowledge that we should dedicate more resources to protecting the data stored. This would involve steps such as encrypting the snapshot images stored on user profiles as well as using a more secure salted-hashing algorithm.

We originally set out to include an escrow system with our payments to guarantee transactions between decorator and customer. This posed several technical hurdles during the development stage, resulting in us switching to a standard PayPal payment page. A full prototype for this was developed and is soon to be utilised in future versions of the application.

Reflecting back on and assessing our development of the chat feature; we struggled to formulate a fully operational system. This was in part due to limited online resources regarding the SendBird API, combined with our inexperience with building networked multi-user tools. Nonetheless, in hindsight we feel we could have achieved a full implementation had we dedicated more group members toward the task.

Regarding the wall colouring, despite our eagerness for computational content, we discovered that this was a slightly over-ambitious step given our resources available. Irrespective of this, we have conducted relevant research on the matter, including subjects such as k-clustering and general image segmentation (19, 20).

We believe our approach to testing was one of the key aspects of delivering a successful project. Each session, whether formal or otherwise allowed us to validate the quality of our current work and offered insight into our developmental priorities. One lesson we took from this testing was how difficult it is to make a fully intuitive app. While most participants didn't struggle, we decided to create a short user guide to clear up any confusion (Appendix E.1). In future we would like to add small pop up tips in-app for new users.

7.3 Future development

In addition to the several planned features discussed throughout this report; we have also considered the potential scalability and drafted plans for future growth as well as sustainability. Four of the long-term targets we have considered are the following:

Integration with AR Hardware

Augmented reality is a booming field within the industry, with several large companies working on their own hardware solutions for a fully immersive experience. The product we have looked into is Microsoft's HoloLens, a pair of mixed reality smart-glasses. While not entirely necessary for the design aspect of our app, tools such as this grant the user a whole new level of interaction. The HoloLens and similar hardware devices will be kept on our radar as the app progresses.

Allowing furniture retailers to sell goods via interiAR

Due to the ease of adding new 3D models into our app, we realised external retailers would be able to sell their furniture through our app by providing the assets. This allows customers to visualise more than just IKEA furniture and would eventually lead up to the option of paying for the furniture through interiAR .This could in turn help multiple stakeholders as it provides customers with more choice while also giving smaller firms a platform to showcase their own furniture providing they can get accurate 3D models of it.

White-labelling our application

White-labelling is the idea of allowing other companies to reskin your software under their brand with a set monetary agreement between both parties. This idea was put forward to us by industry members from Calio (21) who we met at the employer's fair and is something we would strongly consider.

Analysing market trends on furniture and design selections

Finally, we considered the possibility of using interiAR to collect data on which furniture was being commonly selected as well as what styles of design users prefer most. Analysing these market trends could provide extremely useful information to producers of furniture and decorators.

8. Conclusion

To summarise; as a group, we at interiAR believe that we have been considerably ambitious in the undertaking of our project. We have managed to utilise our differing skill-sets, and endeavoured to keep to our development methodology in order to achieve notable progress towards a fully functioning and unique product.

While we haven't perfected every feature we set out to implement, there is a strong base for progression, with several non-trivial functioning components despite unforeseen disruptive circumstances during term outside of our control. The elements we did manage to include have been extensively tested and evaluated both in-house and with several sessions of user testing.

To complement our existing foundations, we have also formulated and outlined substantial improvements and scalability strategies for any future development. This came about via user-feedback, discussion with industry technicians and our own independent research.

Please see a link to our code base hosted on gitLab in appendix E.2.

9. Bibliography

1 Augmented Reality App Development - introduction [Internet]. Waracle. 2016 [cited 16 April 2018]. Available from: https://waracle.com/introduction-augmented-reality-app-development/

2 Unity software [Internet]. Unity3d. 2018 [cited 28 April 2018]. Available from: https://unity3d.com/

3 Radeck K. C# and Java: Comparing Programming Languages [Internet]. Docs.microsoft.com. 2003 [cited 22 April 2018]. Available from: https://docs.microsoft.com/en-us/previous-versions/windows/embedded/ms836794(v=msdn.10)

4 Wikitude Augmented Reality- The World's Leading Cross-Platform AR SDK [Internet]. Wikitude. 2018 [cited 28 April 2018]. Available from: https://www.wikitude.com/

5 Vuforia | Augmented Reality [Internet]. Vuforia.com. 2018 [cited 28 April 2018]. Available from: https://www.vuforia.com/

6 MongoDB Hosting: Database-as-a-Service by mLab [Internet]. mLab. 2018 [cited 28 April 2018]. Available from: https://mlab.com/

7 Győrödi C, Gyorodi R, Pecherle G, Olah A. A Comparative Study: MongoDB vs. MySQL. University of Oradea; 2015.

8 Martin R. Agile software development. Harlow, UK: Pearson; 2014

9 Trello [Internet]. Trello.com. 2018 [cited 26 April 2018]. Available from: https://trello.com/

10 Principles behind the Agile Manifesto [Internet]. Agilemanifesto.org. [cited 22 April 2018]. Available from: http://agilemanifesto.org/principles.html

11 Lean Touch - Asset Store [Internet]. Assetstore.unity.com. 2018 [cited 29 April 2018]. Available from: https://assetstore.unity.com/packages/tools/input-management/lean-touch-30111

12 OAuth 2.0 — OAuth [Internet]. Oauth.net. 2018 [cited 26 April 2018]. Available from: https://oauth.net/2/

13 Messaging SDK and Chat API for Mobile Apps and Websites | SendBird [Internet]. SendBird. 2018 [cited 28 April 2018]. Available from: http://sendbird.com

14 Data Protection Act 1998 [Internet]. Legislation.gov.uk. 1998 [cited 24 April 2018]. Available from: https://www.legislation.gov.uk/ukpga/1998/29

15 General Data Protection Regulation (GDPR) – Final text neatly arranged [Internet]. General Data Protection Regulation (GDPR). 2018 [cited 24 April 2018]. Available from: https://gdpr-info.eu/

16 J. Lim J, Pirsiavash H, Torralba A. Parsing IKEA Objects: Fine Pose Estimation. ICCV; 2013.

17 Zion Research - With 85.2% CAGR, Augmented Reality (AR) Market Size Will Exceed USD 133.78 Bn by 2021: Zion Market Research [Internet]. 2018 [cited 16 April 2018].

18 IKEA T, release P, results I. IKEA UK business results - IKEA [Internet]. Ikea.com. 2016 [cited 26 April 2018]. Available from: https://www.ikea.com/gb/en/this-is-ikea/newsroom/press-release/ikea-uk-business-results/

19 Barghout L, Sheynin J. Real-world scene perception and perceptual organization: Lessons from Computer Vision. Journal of Vision. 2013;13(9):709-709.

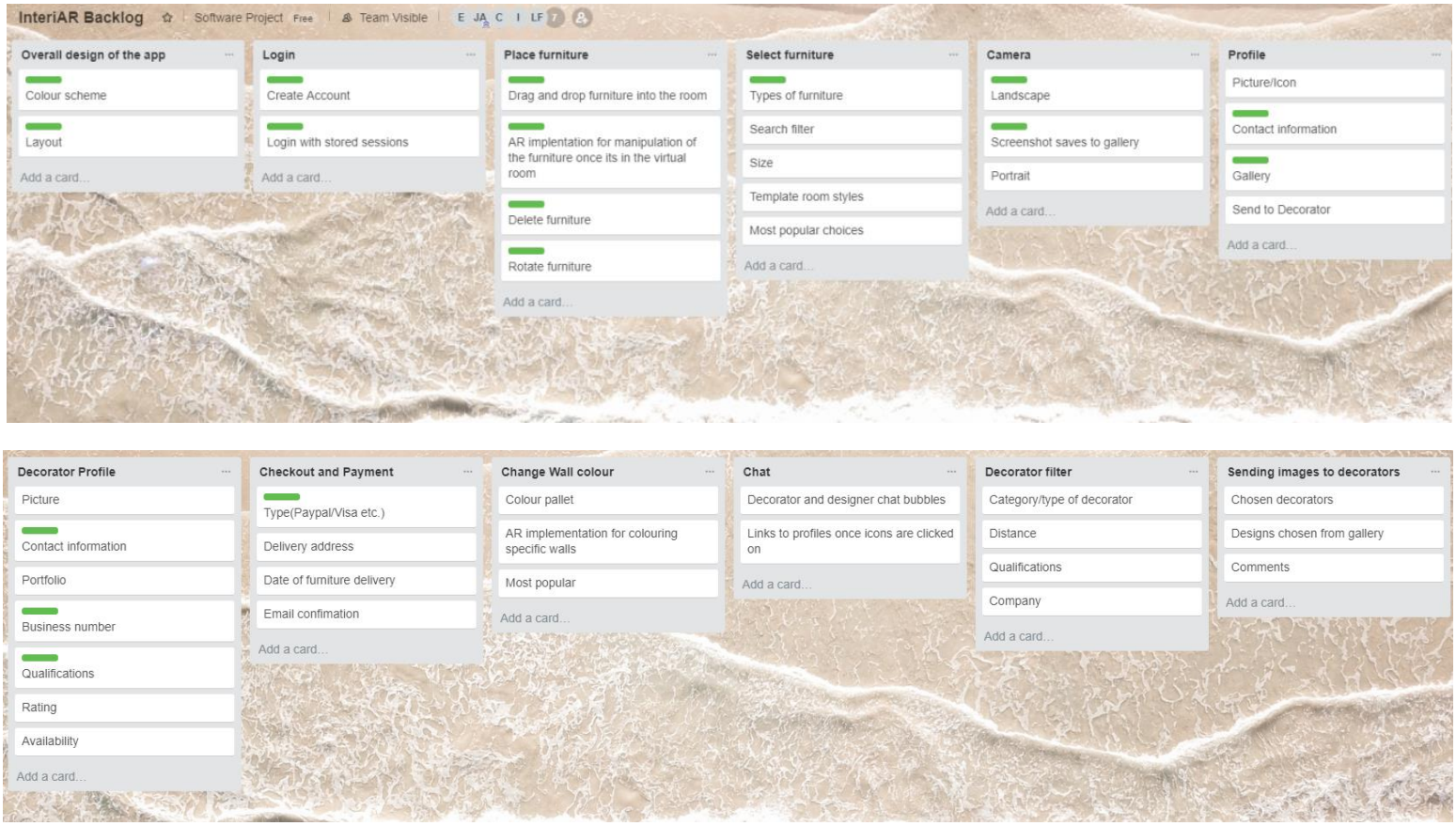
20 Shapiro L, Stockman G. Computer vision. New Jersey, Upper Saddle River: Prentice Hall; 2001

21 Calio | A Really Good Calendar [Internet]. Calio | A Really Good Calendar. 2018 [cited 29 April 2018]. Available from: https://www.calio.co.uk/

10. Appendices

10.1 Appendix A: Development Record

A.1 Overall product backlog



The initial overall product backlog, where a green label indicates being part of the minimum viable product.

A.2 Gantt chart

Gantt chart displaying our milestones and tasks throughout the entire project, spread across two pages for readability

A.3 User stories

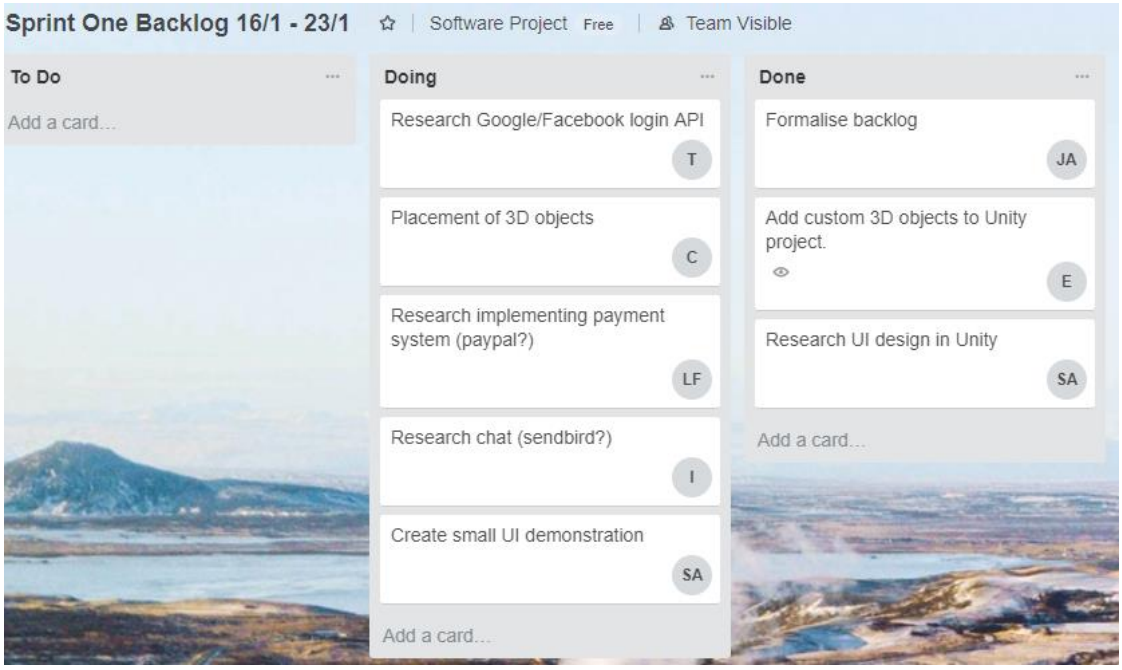
"As a home designer struggling with new ideas, I want to be able to show my clients multiple design ideas with different pieces of furniture, so that my home design work can improve and I can attract new clients with this technology."

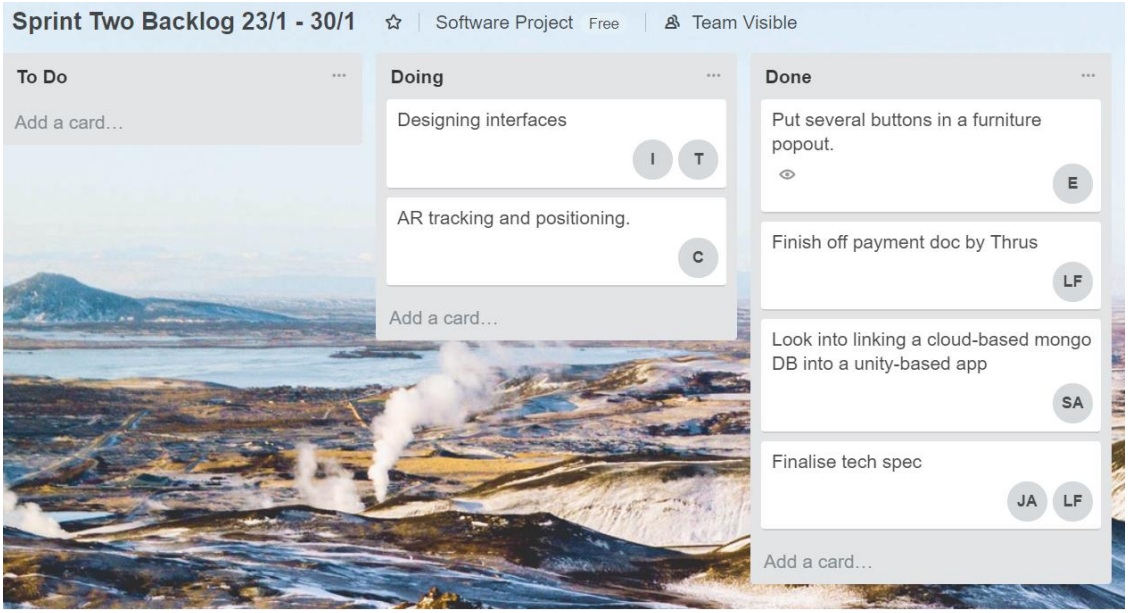
"As a retired home owner, I would like to be able to take and share pictures of my designs with my friends and family, so that I can show off my dream house!"

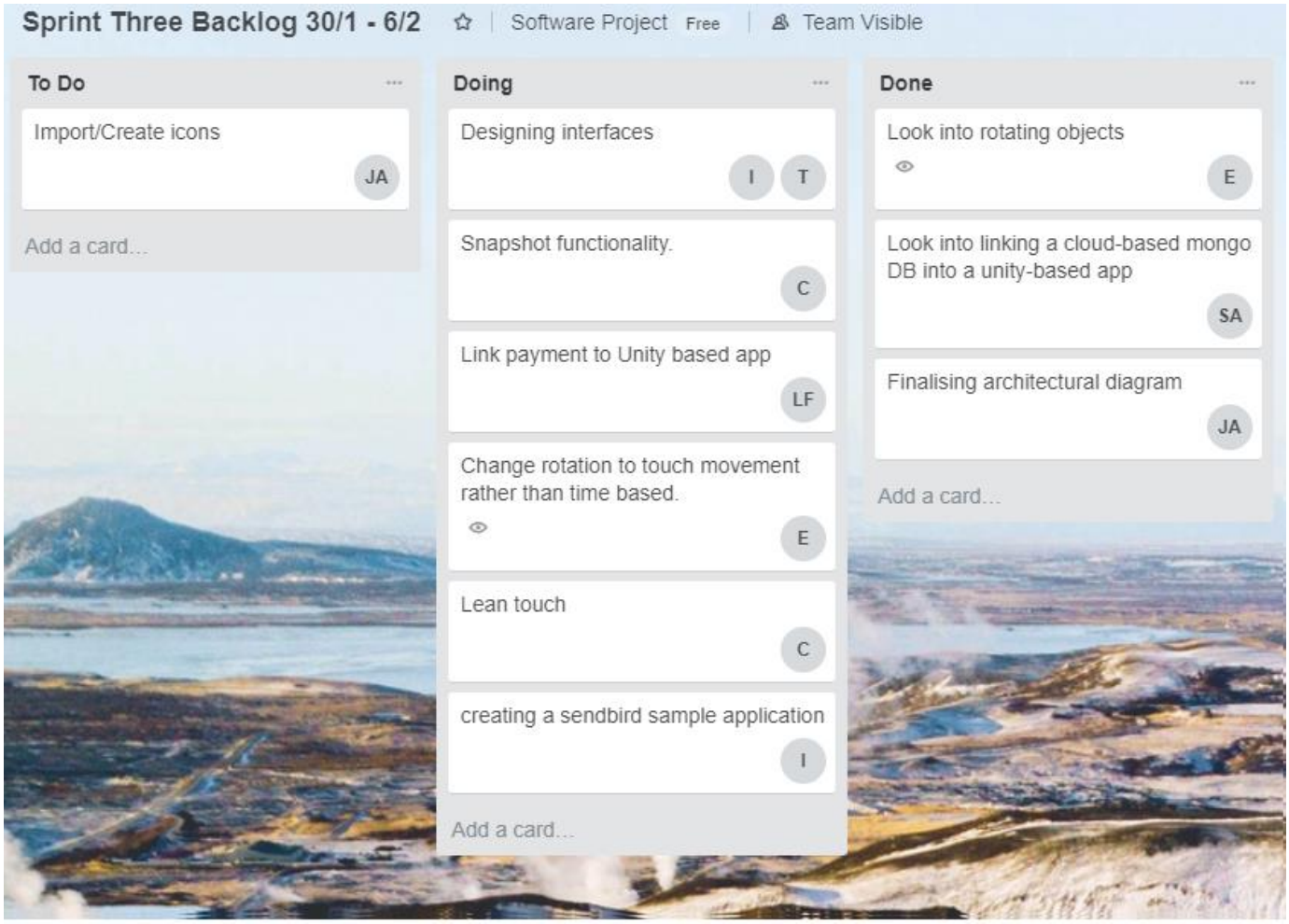
"As a young doctor living a hectic lifestyle in London, I want an app that allows me to easily communicate with decorators, as I hate the constant back and forth as well as have limited free time."

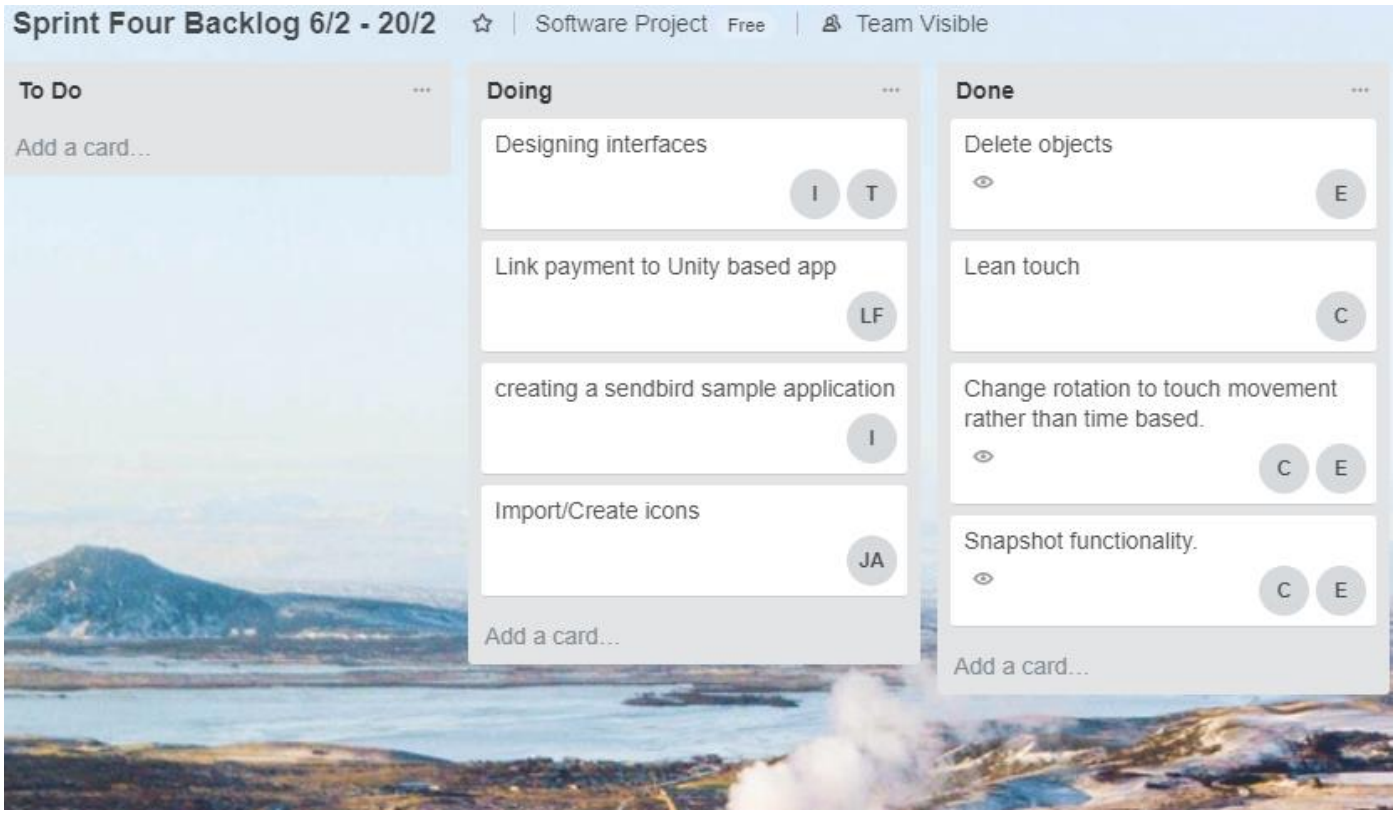
User stories in the early development stages and attempted to fulfil.

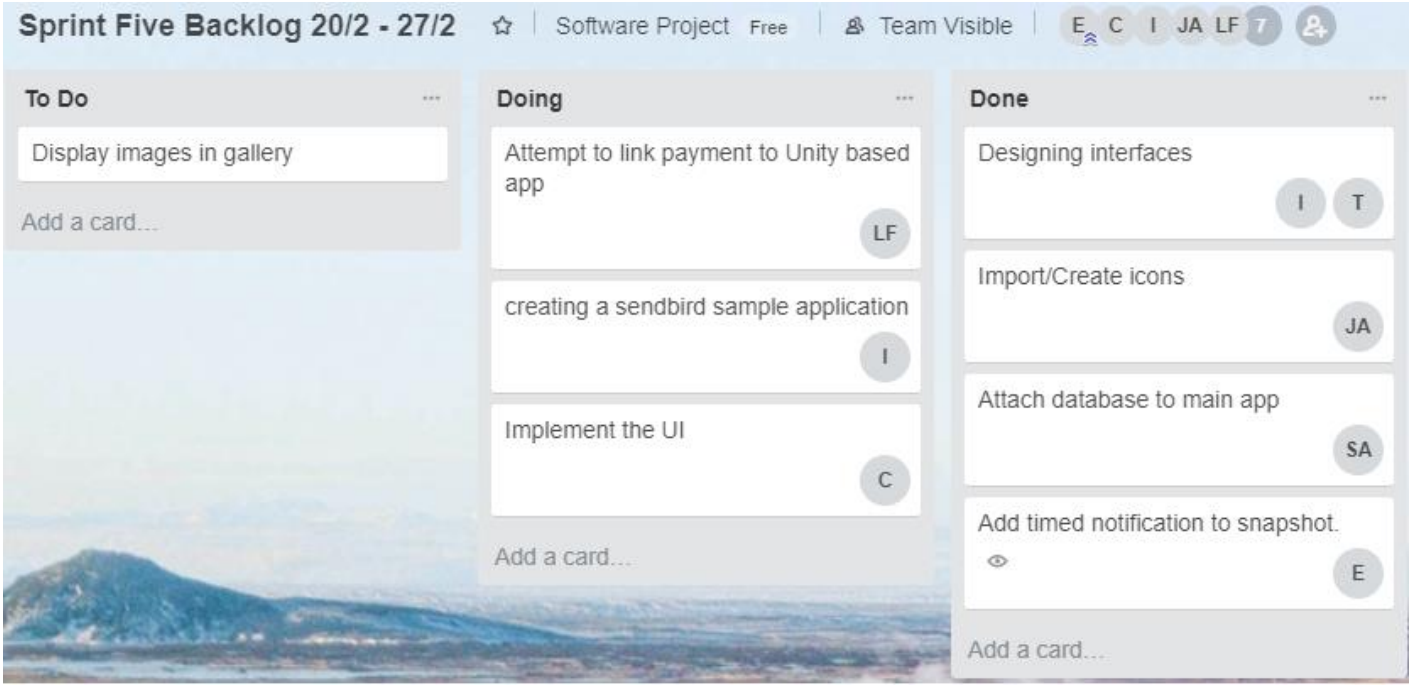
A.4 Sprint backlogs






Sprint Three Backlog 30/1 - 6/2
Software Project Free
Team Visible
To Do
Import/Create icons
JA
Add a card...
Doing
Designing interfaces
I T
Snapshot functionality.
C
Link payment to Unity based app
LF
Change rotation to touch movement rather than time based.
E
Lean touch
C
creating a sendbird sample application
I
Add a card...
Done
Look into rotating objects
E
Look into linking a cloud-based mongo DB into a unity-based app
SA
Finalising architectural diagram
JA
Add a card...


Sprint Four Backlog 6/2 - 20/2
Software Project Free
Team Visible
To Do
Add a card...
Doing
Designing interfaces
I T
Link payment to Unity based app
LF
creating a sendbird sample application
I
Import/Create icons
JA
Add a card...
Done
Delete objects
E
Lean touch
C
Change rotation to touch movement rather than time based.
C E
Snapshot functionality.
C E
Add a card...



Screenshots of individual sprint backlogs taken weekly. Included the first half-term.

A.5 Scrum notes

18/01/2018 Scrum InteriAR Group H
Tanzum: Compiled word doc on Google API information, links to guides for use on mobile. Facebook to be done tomorrow.
Ethan: Created a mini prototype of a pre-build instant tracking example and imported in objects from our database of IKEA models.
Jibril: Compiled a detailed Trello-based product backlog.
Liban: Gathered information on Paypal/Mastercard, compiling word doc for tomorrow.
Ifrah: Looked into how SendBird works – will compile word doc for next scrum.
Shah: Absent
Cleon: Absent

19/01/18 Scrum Meeting
Ethan: Added a secondary object to the prototype, will look into the code elements of buttons etc.
Cleon: Started testing instant tracking with Wikitude will try get working in-app and look at grid for Tuesday.
Jibril: Finished off the backlog will get to grips with Unity for Tuesday.
Ifrah: In process of compiling the word on SendBird, will finish for Tuesday setup Unity and Wikitude.
Shah: Has looked into UI, created a few buttons. Will have a demonstration app by Tuesday.
Tanzum: Absent
Liban: Absent

23/01/18 Scrum
Ifrah: Completed and uploaded brief doc on SendBird chat API.
Tanzum: Completed and uploaded brief doc on Google/Facebook login API's.
Cleon: Was struggling to fix building to his iPhone. Now fixed.
Liban: Finishing off his doc, done by thurs.
Ethan: Added test buttons/elements looked more at documentation.
Jibril: Set up Unity with Wikitude, getting familiar with it.
Shah: Built a small demo of UI within Unity.

25/01/18
Ethan: Has added a pop out menu with the furniture buttons.
Cleon: Has been struggling to fix license key for Wikitude, causing delay on development.
Tanzum: Trying to sort Photoshop.
Ifrah: Also trying to sort Photoshop
Jibril: Sorting out android SDK's and has made a start on the spec.
Liban: Made a start on spec.
Shah: Absent

26/01/18
Ethan: Minor changes to the UI buttons, not much since yesterday.
Cleon: Still trying to fix licensing, currently in contact with Wikitude. (fixed after meeting – he messed up a capital letter the absolute wally).
Jibril: Has successfully deployed an example app to Android.
Liban: Has been working on the tech specification, making progress for documentation.
Ifrah: Has Photoshop set up.
Tanzum: Absent
Shah: Absent but has done some work on database side

30/01/18
Ethan: Sorted out orientation of the app, left-landscape.
Ifrah: Been looking into logo design and learning photoshop.
Tanzum: Been looking into logo design and colour scheme.
Jibril: Finished tech spec.
Liban: Finished tech spec. Will work on more docs.
Shah: Has found and been trying to test a MongoDB driver for Unity, unfortunately there's only a legacy driver available as Unity uses an older framework.
Cleon: Absent

01/02/18
Ethan: Got objects rotating on 3 touch gesture, but only time based not movement.
Tanzum: Been playing around with Photoshop, focusing on logo.
Liban: Looking into payment implementation with Unity.
Jibril: Sorting diagrams.
Ifrah: Focusing on UI.
Cleon: Looked into lean touch. Will now also look into snapshots.
Shah: Absent

02/02/18
Cleon: Has found a way to sort snapshots, will be implemented on a test app by Tuesday.
Ethan: Added code to hide menu on object select, fixed box collider.
Liban: Payment system for unity still in progress.
Jibril: Will have tech spec finished today.
Shah: Working with C# on database, managed to insert document, will look at displaying based on database.
Ifrah: Working on logo, developing
Tanzum: Absent

06/02/18
Ethan: Working on menus/submenus functionality of furniture.
Cleon: Been working on snapshot functionality, continuing into this sprint.
Ifrah: Looking at implementing SendBird into a sample application and has a mock up of a logo.
Shah: Has been able to insert documents via a unity app, moving onto displaying and creating an actual mock-up of decorator profile form.
Jibril: Absent
Liban: Absent
Tanzum: Absent

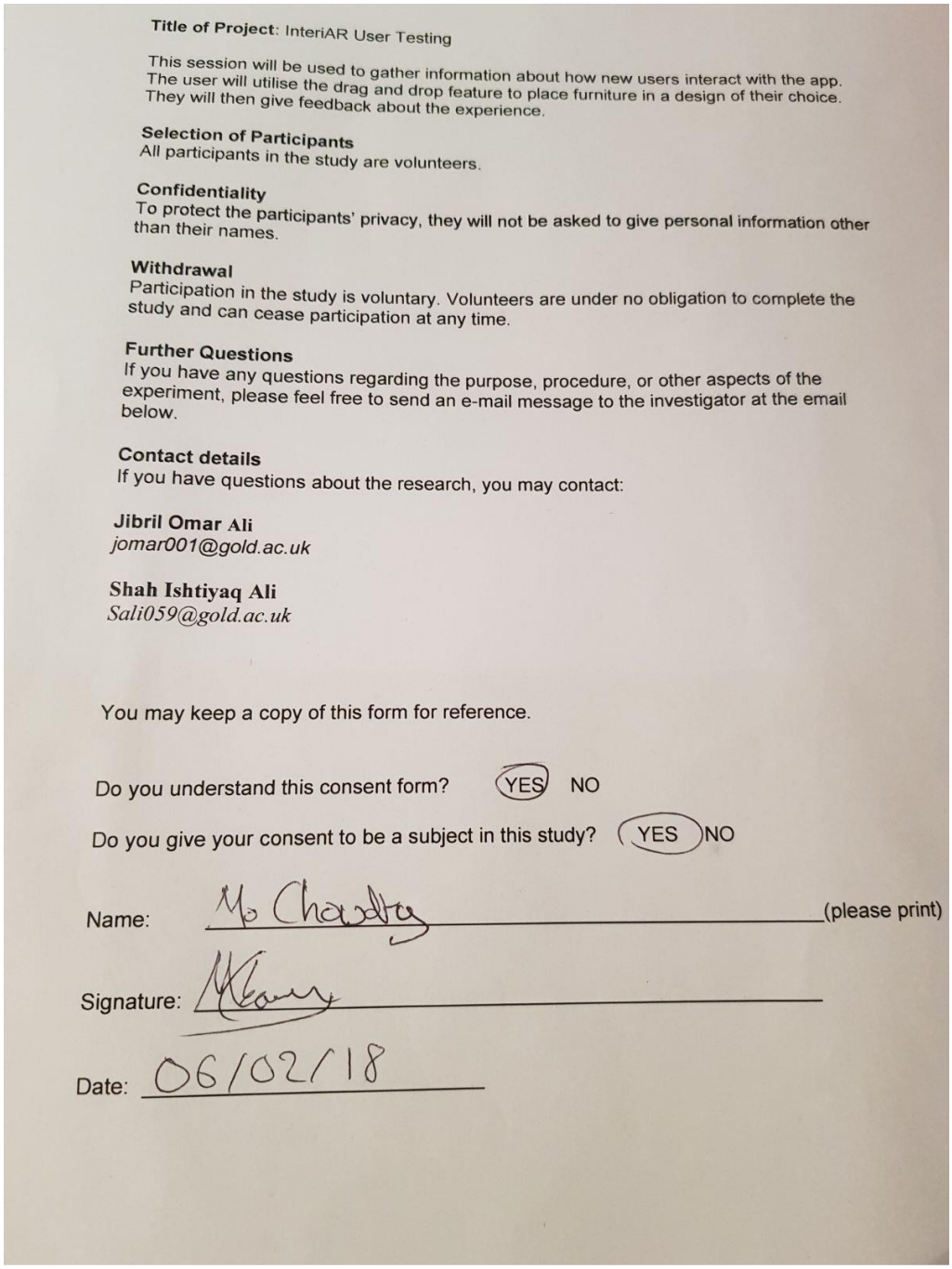
08/02/18
Ethan: Got 3 subsections, displaying 3 examples of that type of furniture each.
Tanzum: Has drawn up some interfaces designs.
Jibril: Finished diagram and wrote up some data gathered from very early user testing.
Liban: Has started to code up a payment system, in java at the moment, will try link to unity layer or look into possibilities as it isn't straight forward.
Shah: Absent
Ifrah: Absent
Cleon: Absent

09/02/2018
Ethan: Minor tweaks from yesterday, cleaned up furniture buttons layout.
Jibril: Working on adding custom icons to unity.
Ifrah: Still looking into chat with sendbird
Shah: Looking into login, found google api requires payment and might be complicated – going to look into backup options while exploring google further.
Liban: Still working on payment stuff, working in java but will look into a way to combine with unity.
Cleon: Absent
Tanzum: Absent

Brief notes taken in scrum meetings. Included the first half-term only.

10.2 Appendix B: Formative Evaluation

B.1 User consent form



Title of Project: InteriAR User Testing

This session will be used to gather information about how new users interact with the app. The user will utilise the drag and drop feature to place furniture in a design of their choice. They will then give feedback about the experience.

Selection of Participants
All participants in the study are volunteers.

Confidentiality
To protect the participants' privacy, they will not be asked to give personal information other than their names.

Withdrawal
Participation in the study is voluntary. Volunteers are under no obligation to complete the study and can cease participation at any time.

Further Questions
If you have any questions regarding the purpose, procedure, or other aspects of the experiment, please feel free to send an e-mail message to the investigator at the email below.

Contact details
If you have questions about the research, you may contact:

Jibril Omar Ali
jomar001@gold.ac.uk

Shah Ishtiyaq Ali
Sali059@gold.ac.uk

You may keep a copy of this form for reference.

Do you understand this consent form? (YES) NO

Do you give your consent to be a subject in this study? (YES) NO

Name: Mo Chowdhury (please print)

Signature:

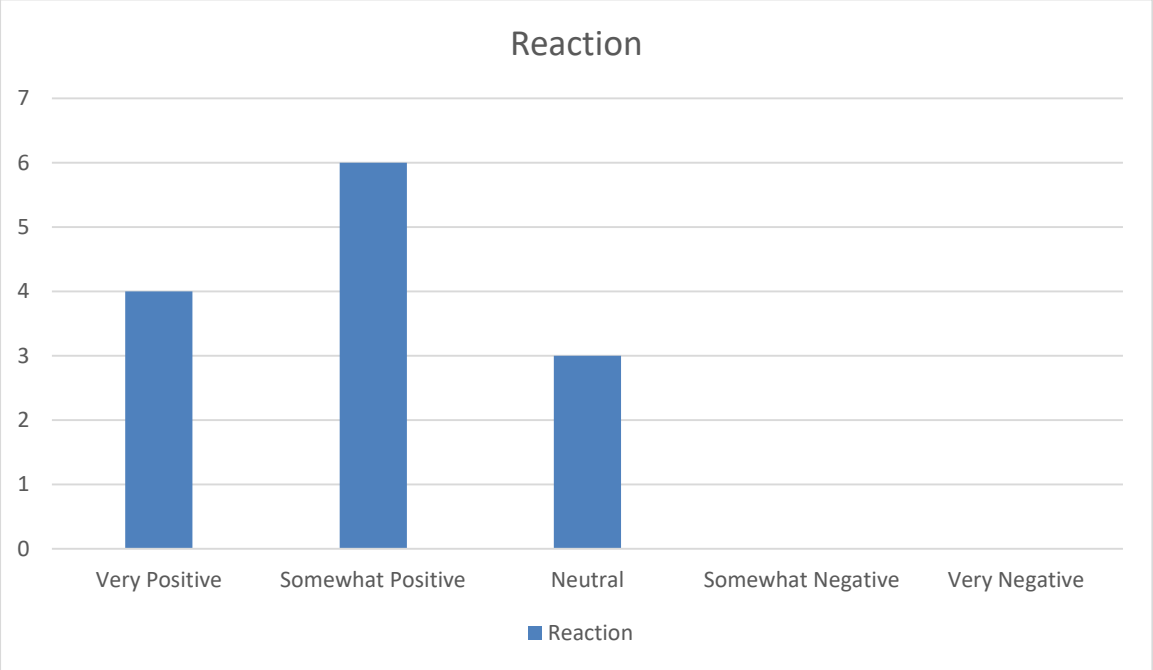
Date: 06/02/18

Example of one of our user consent forms signed by a participant. The rest of the forms are available upon request.

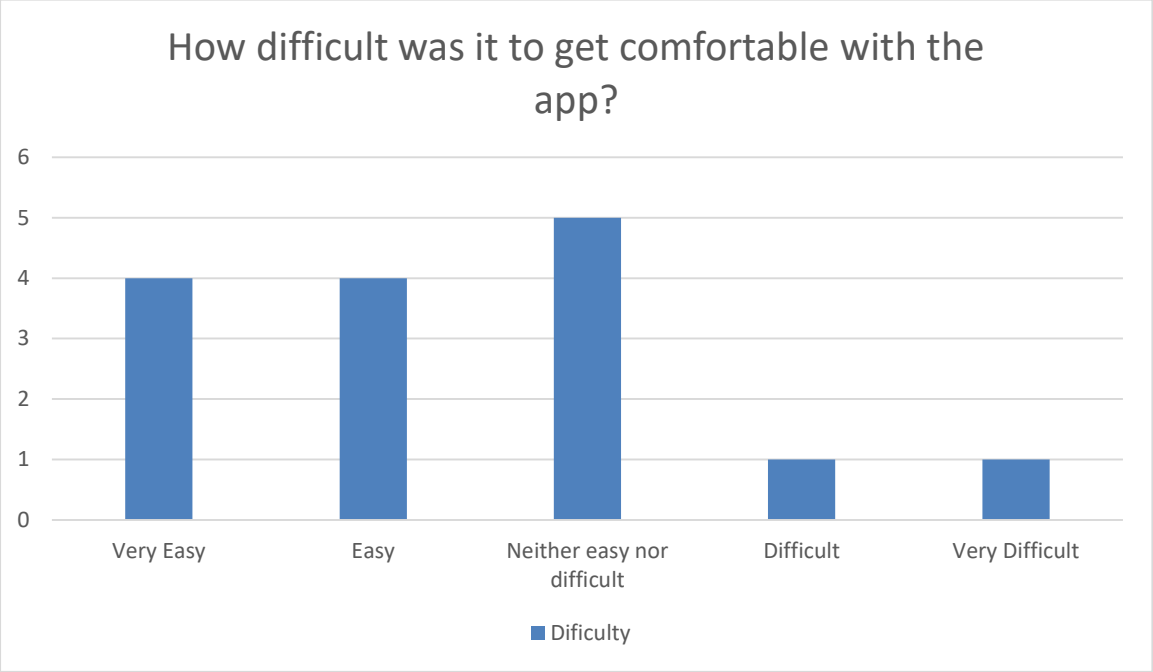
B.2 AR camera user testing

Survey Questions:

1	What was your first reaction to the app?



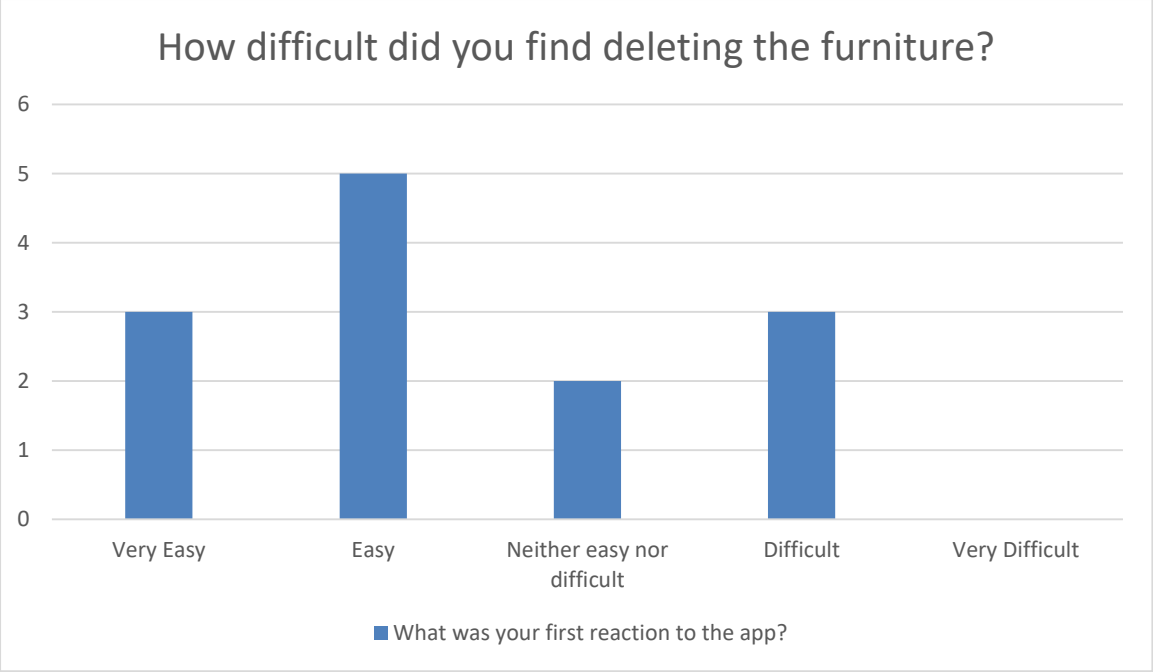
2	How difficult was it to get comfortable with the app?



3	What do you think about the user interface?

Users	Responses
User 1	It's simple and easy to understand for the most part
User 2	A little too simplistic, doesn't include many different features
User 3	The icons are a bit large for the screen
User 4	The icons look too cartoony
User 5	Screen looks messy
User 6	The icons should look more realistic
User 7	Text is difficult to see
User 8	The picture of the furniture should be the icons
User 9	The slider on the right is confusing
User 10	Its simple to use and understand
User 11	No clear guidelines on how to use the interface
User 12	Back button too obscure
User 13	All the buttons are very responsive

4	How difficult did you find deleting the furniture?



5	What do you like the most about the apps features?

Users	Responses
User 1	Easy to drag, drop and reposition the furniture
User 2	Easy to position once it's on the screen
User 3	The furniture manages to stay in place even when I look away.
User 4	Extremely easy to visualise potential design ideas
User 5	The furniture looks proportionate to real furniture
User 6	Not complicated to navigate
User 7	The screen was uncluttered which meant that it was easy to view the room and make changes
User 8	The screen isn't very complex
User 9	Gives me an outlet to be creative
User 10	There aren't many functions around the screen, so it allows more of the screen to be free for viewing my design
User 11	Furniture stays on the floor
User 12	That the furniture has been taken directly from the ikea catalogue
User 13	The app doesn't lag when I place the objects

6	What are the things that you would want to see improved?

Users	Responses
User 1	It took me a few goes to figure it out, a help section might be useful
User 2	Design
User 3	The ability to rotate the furniture
User 4	The back button should remove all of the furniture, just the last one
User 5	Kept double tapping the furniture icons when trying to drag them in and ended up making multiple unwanted furniture items, and had to get help with taking them off, would be useful for a way to delete items individually.
User 6	The text buttons look a little boring, doesn't look very appealing, would rather have a simple furniture icon.
User 7	More variety of furniture
User 8	Text tends to blend into bright backgrounds
User 9	Different catalogues to choose from(e.g. dfs)
User 10	Clearer way of deleting items, one by one rather than resetting the whole design (which really is a frustrating)
User 11	Categorise more items into their own type
User 12	Provide the measurements for the furniture
User 13	Make the back button have and 'undo' feature instead

7	Which features would you want implemented in the future?

Users	Responses
User 1	Rotation
User 2	Add a tutorial option to walk through new users around the app
User 3	Implement a furniture rotation feature
User 4	Rotation feature
User 5	A large library of furniture to choose from
User 6	Deeper trees of furniture e.g chair button includes armchairs as well as High chairs
User 7	Include prices of the furniture
User 8	Panoramic views
User 9	Add a menu bar to make it less cluttered
User 10	Options to change the colour of the furniture
User 11	Option to alternate the speed of the drag and drop movement
User 12	Have a screenshot feature that saves the page to a gallery
User 13	Ability to rotate the furniture

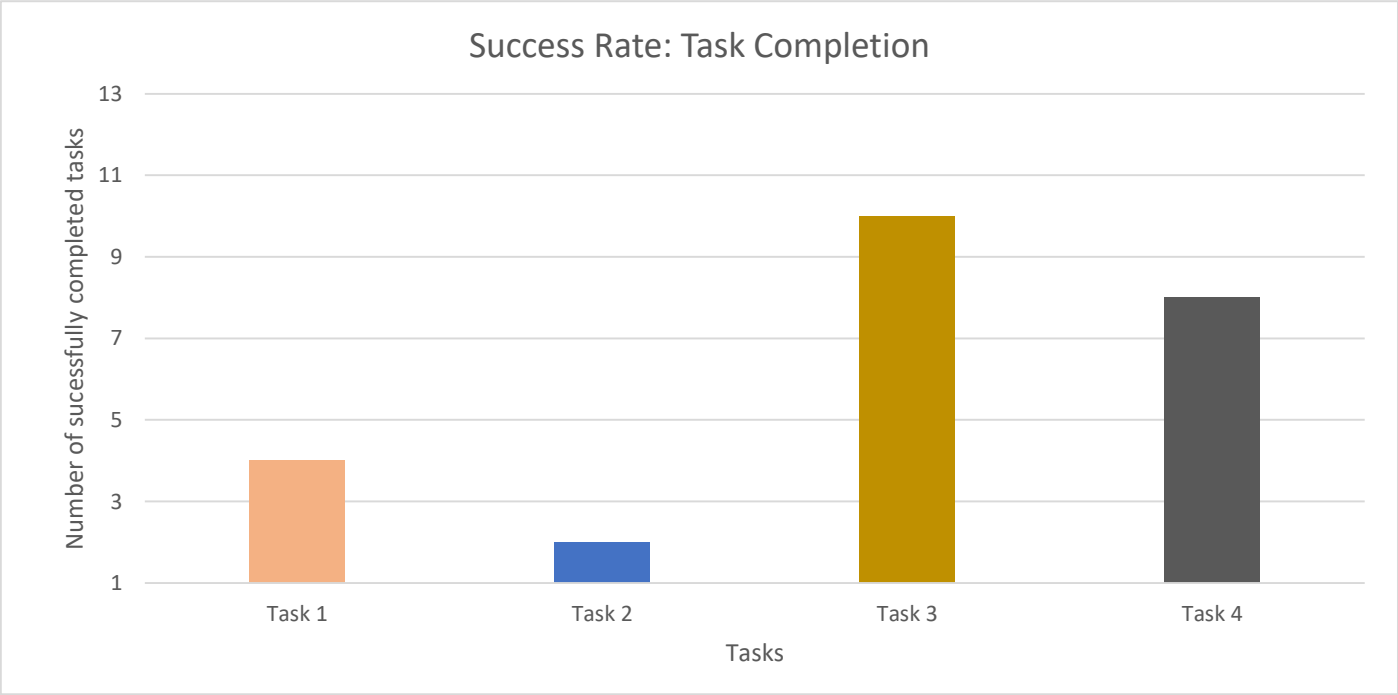
B.3 Login and profile user testing

Task 1	On your first or second try, were you able to create an account successfully?
Task 2	How clear was the login screen in specifying you were inputting the wrong username/password?
Task 3	Has your profile page displayed incorrect information after editing/updating profile?
Task 4	Were you able to open the AR (augmented-reality) camera through the camera button on your first try?

All the tasks our 13 users performed

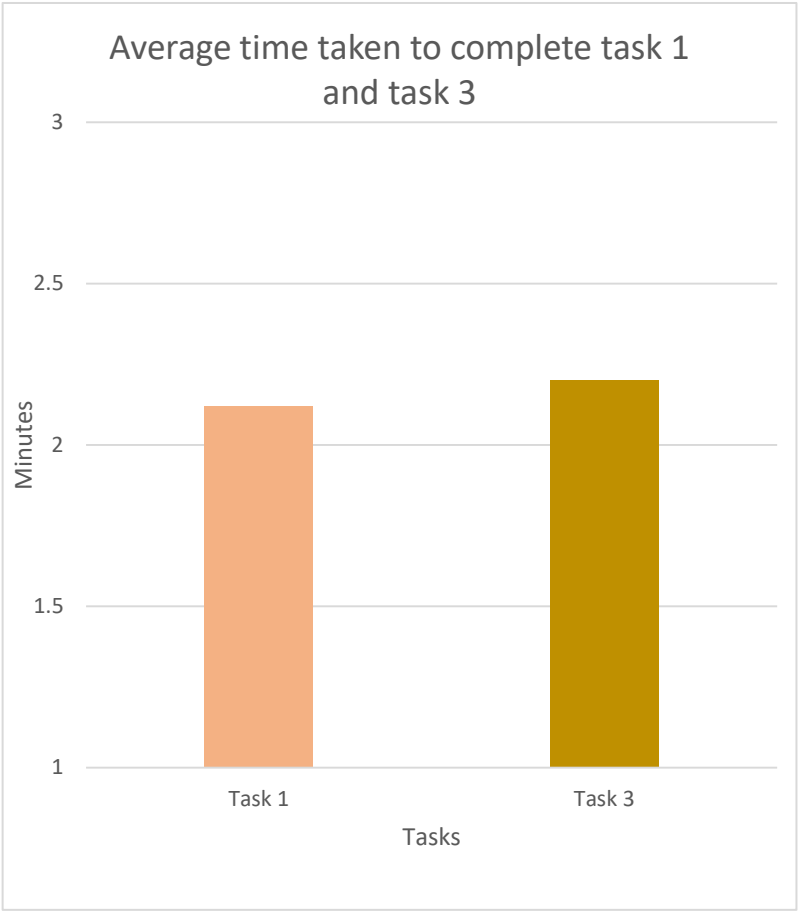
The results users provided to us for each task

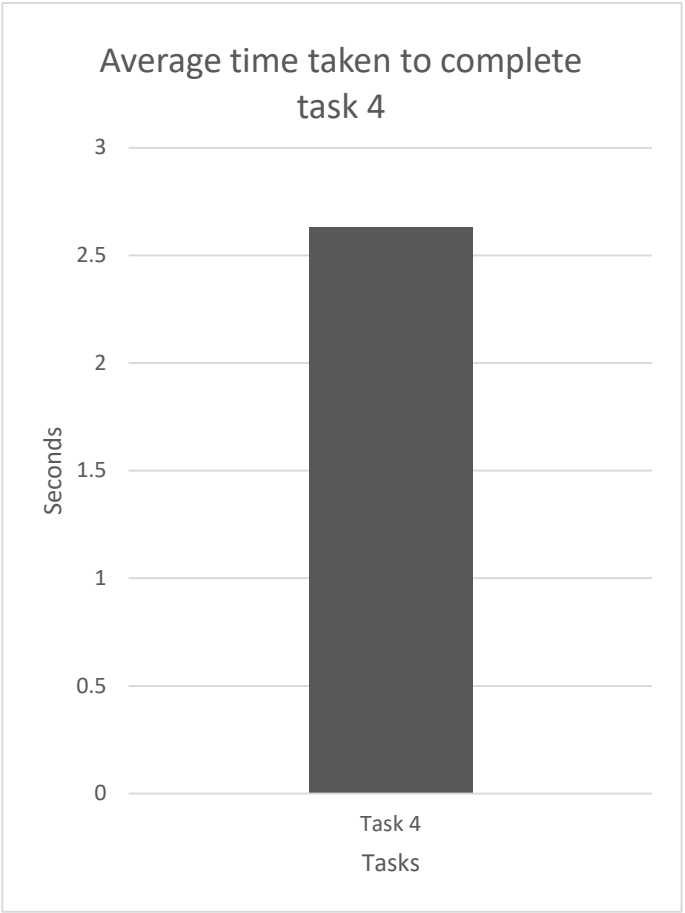
	USERS ONLY			
Our Users	Task 1	Task 2	Task 3	Task 4
User 1	No	Unclear	No	No
User 2	No	Unclear	No	Yes
User 3	No	Clear	No	Yes
User 4	No	Unclear	No	No
User 5	No	Unclear	No	Yes
User 6	Yes	Unclear	No	Yes
User 7	No	Don't know	No	No
User 8	Yes	Unclear	Don't know	Yes
User 9	No	Unclear	No	Yes
User 10	Yes	Clear	No	No
User 11	No	Unclear	Don't know	No
User 12	Yes	Don't know	No	Yes
User 13	No	Unclear	Don't know	Yes



From the bar chart, it is easy to tell that users struggled to complete task 1 and task 2. It seems that users are finding it difficult to create an account and understand the login process. This is something that came up on our software testing (non-functional – usability testing.) We found that the create account screen was not user-friendly as users had to input too much. And the login screen was unclear in specifying when users were inputting an invalid username or password as it didn't display any form of error message. This is something we manage to clear up in our latest version. Tasks 3 and 4 seem to be completed successfully by most of our users.

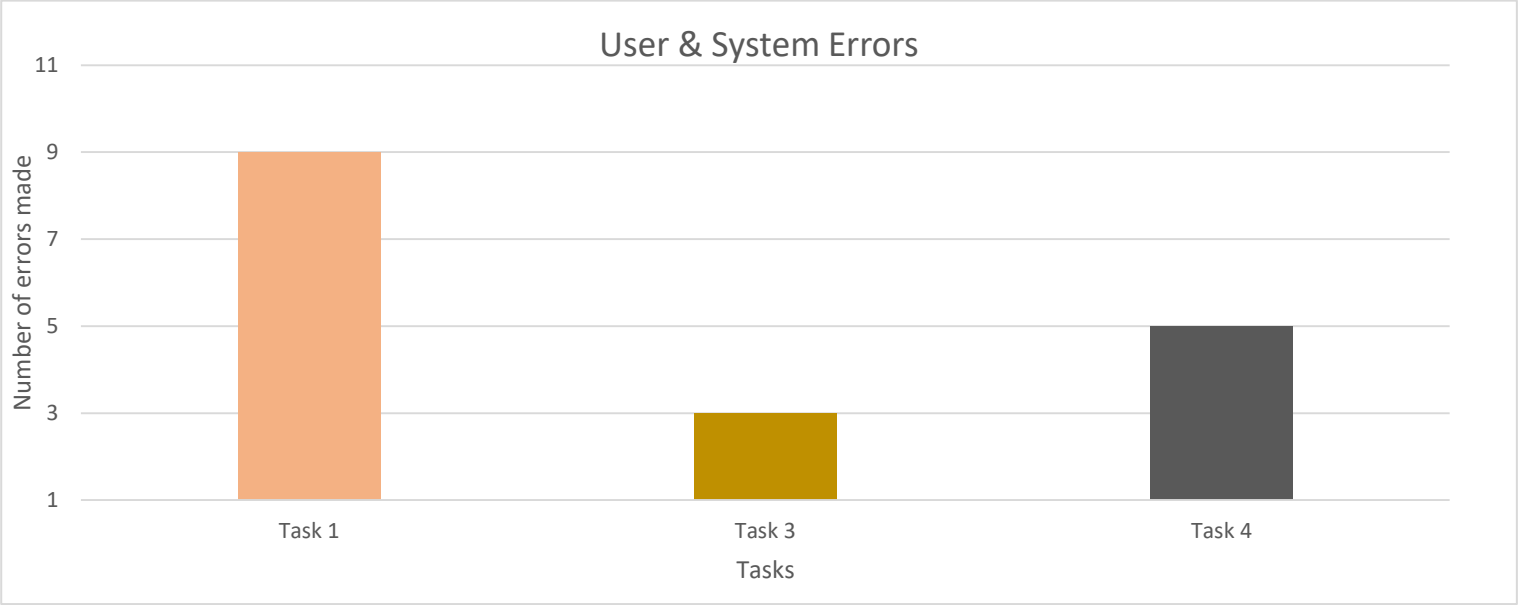
	DEVELOPERS ONLY		
Our Users	Task 1	Task 3	Task 4
User 1	2 - 3 mins	1 - 2 mins	N/A
User 2	1 - 2 mins	1 - 2 mins	1 - 3 seconds
User 3	2 - 3 mins	2 - 3 mins	1 - 5 seconds
User 4	2 - 3 mins	2 - 3 mins	N/A
User 5	1 - 2 mins	2 - 3 mins	1 - 3 seconds
User 6	0 - 1 mins	2 - 3 mins	1 - 5 seconds
User 7	0 - 1 mins	3 - 4 mins	N/A
User 8	2 - 3 mins	N/A	1 - 5 seconds
User 9	3 - 4 mins	2 - 3 mins	1 - 5 seconds
User 10	2 - 3 mins	1 - 2 mins	N/A
User 11	1 - 2 mins	N/A	N/A
User 12	2 - 3 mins	1 - 2 mins	1 - 5 seconds
User 13	3 - 4 mins	N/A	1 - 3 seconds





Despite users having issues completing tasks 1 and tasks 2, the bar chart shows a decent average time that users took to complete them. However, most of our users typed in anything within each of the input fields. If users were to input with a bit more awareness, then maybe the average time could've been higher. But right now it is difficult to predict. The average time it took to complete task 4 is also decent. On average it takes 2.63 seconds for the application to open the AR camera. This result seems like the best we've received so far, since AR cameras are highly advanced and take a lot of memory space and for it to load at an average time of 2.63 seconds is pretty impressive. Task 2 was not recorded as this was more of a measure of the perspective of our users.

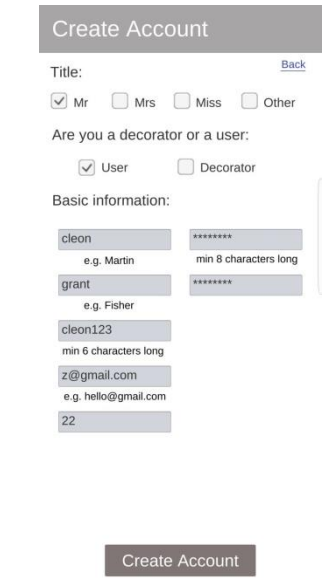
	DEVELOPERS ONLY		
Our Users	Task 1	Task 3	Task 4
User 1	Error on email field	N/A	Displayed a could not load error message
User 2	Error on password field	N/A	N/A
User 3	Error on DOB fields	N/A	N/A
User 4	Error on username field	N/A	Displayed a could not load error message
User 5	Error on password field	N/A	N/A
User 6	N/A	N/A	N/A
User 7	Error on username field	N/A	Displayed a could not load error message
User 8	N/A	Error in the edit profile page: on password field	N/A
User 9	Error on password field	N/A	N/A
User 10	N/A	N/A	Displayed black empty screen
User 11	Error on password field	Error in the edit profile page: Error on email field	Displayed a could not load error message
User 12	N/A	N/A	N/A
User 13	Error on password field	Error in the edit profile page: Error on password field	N/A



From the bar chart, it looks like a lot of errors were made in task 1. This seemed only predictable, since we previously analysed that the create account screen was not user-friendly i.e. required too much user input. Likewise, for task 3 where the same interface was used within the edit profile page. This, according to our table is where all the errors were made. For task 4 a few users had issues loading the AR camera. In most cases, it seemed like a rendering issue. This is something that can easily be fixed in Unity via the player settings. But, most users were able to open the AR camera repeatedly without any errors.

10.3 Appendix C: Design and Implementation

C.1 Login and account creation



This is the sign up page all users have to go through to create an account. There are several fields that have to be filled in and validated before the account is created.

Conceptual

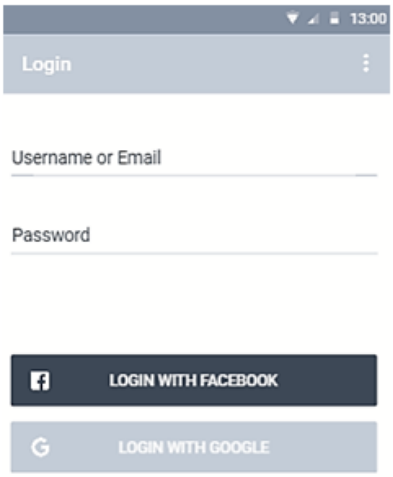


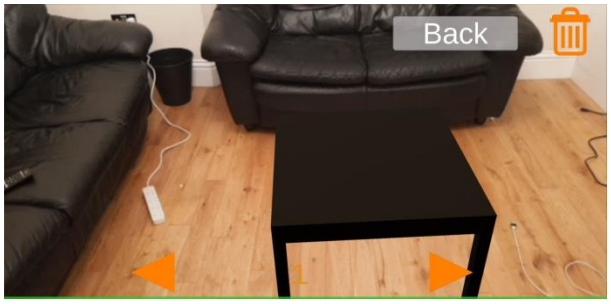
Fig 1

Current Version

Fig2

According to the design of our conceptual prototype, we were show our intent on using Facebook and Google to authenticate our users shown in fig.1
Fig.2 shows that we have switched to using our own login page, the dots is the system checking the credentials against the mongo database.

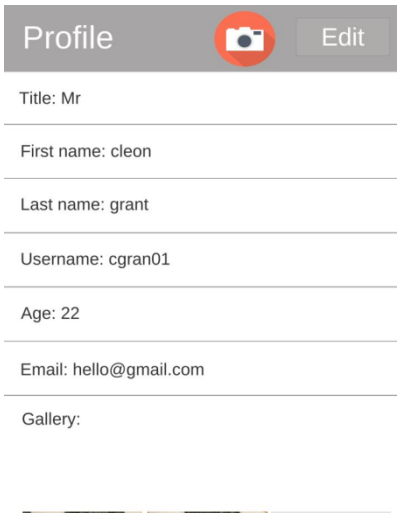
C.2 Snapshot and Profiles




```
using UnityEngine;
using System.Collections;
using UnityEngine.SceneManagement;

//put non capture stuff in a diff class.
using MongoDB.Driver;
using System.Security.Cryptography;
using System.Text;
using MongoDB.Bson;
using MongoDB.Driver.Builders;

 class Snapshot : MonoBehaviour {
    private int counter = 0;

    //notifcation
    public GameObject notification;

    public void capture(){
        //assigning popup object
        //notification = GameObject.Find("SnapshotPopup");

        //Making the UI invisible
        GameObject.Find("SnapshotButton").transform.localScale = new Vector3(0, 0, 0);
        GameObject.Find("DeleteButton").transform.localScale = new Vector3(0, 0, 0);
        GameObject.Find("BackButton").transform.localScale = new Vector3(0, 0, 0);
        GameObject.Find ("Reinitialise").transform.localScale = new Vector3 (0, 0, 0);
        GameObject.Find("Furniture").transform.localScale = new Vector3(0, 0, 0);

        //Giving the SS a name based on date/time
        counter++;
        string file_name = "Screenshot_" + counter + ".jpg";
        //Saving this SS to the device.
        ScreenCapture.CaptureScreenshot (file_name);
        //As CaptureScreenshot is asynchronus, a delay must be used to make sure the UI isn't made visible before the SS happens.
        StartCoroutine (ssDelay ());

        StartCoroutine (ssNotification ());

    }



    public void addimgtoDB() {
        var name = "ishtiyaq";
        var pass = "1234";
        var dbname = "decorators";
        var uri = "mongodb://" + name + ":" + pass + "@ds119268.mlab.com:19268/decorators";

        MongoClient client = new MongoClient(uri);
```

```
        if (counter == 6) {
            var fileName = Application.persistentDataPath + "/" + "Screenshot_6.jpg";

            collection.Update (Query.And (
                Query.EQ ("username", PlayerPrefs.GetString ("unme"))
            ), Update<fields2>.
                Set (s => s.directory6, fileName));
        }


    }

    private IEnumerator ssDelay(){
        yield return 0;     //waits for next frame
        //Making the UI visible again.
        GameObject.Find("SnapshotButton").transform.localScale = new Vector3(0.1f, 0.5f, 1);
        GameObject.Find("DeleteButton").transform.localScale = new Vector3(0.1f, 0.5f, 1);
        GameObject.Find("BackButton").transform.localScale = new Vector3(1, 1, 1);
        GameObject.Find("Furniture").transform.localScale = new Vector3(1, 1, 1);
        GameObject.Find("Reinitialise").transform.localScale = new Vector3(1, 1, 1);
    }

    private IEnumerator ssNotification(){
        yield return 0;
        notification.SetActive(true);
        yield return new WaitForSeconds (3);
        notification.SetActive(false);
    }
```

Use of IEnumerators above deals with the asynchronous nature of the capture screenshot method.

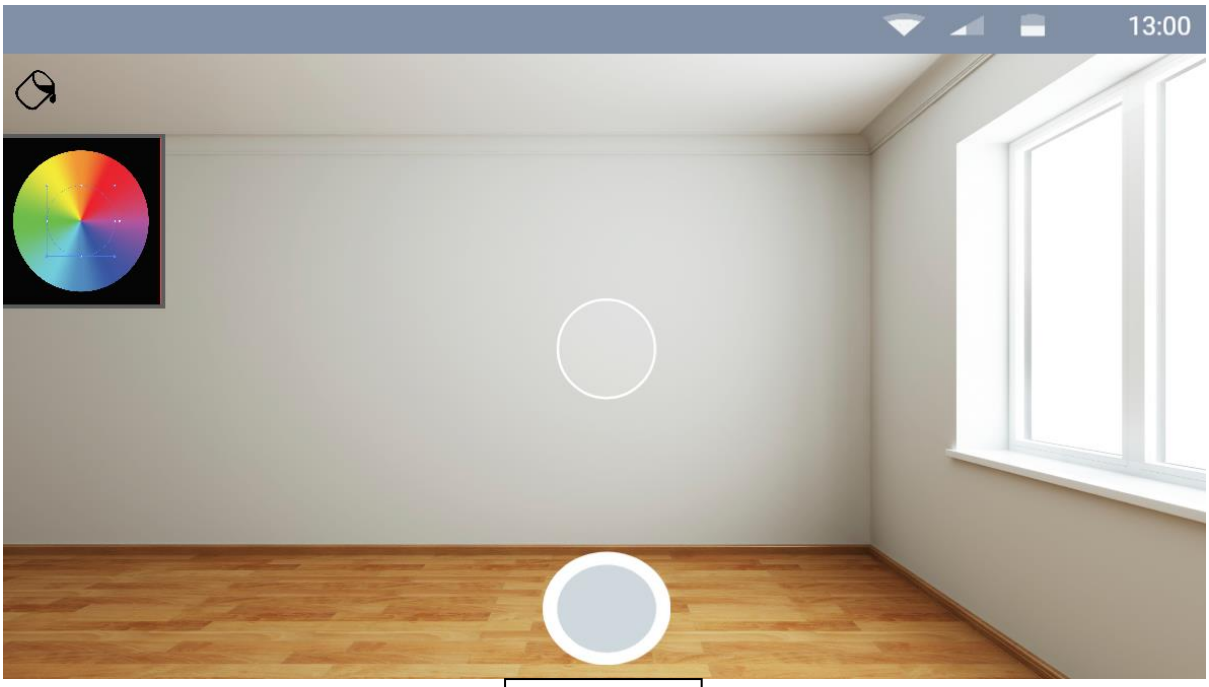


Profile screen

```
MongoClient client = new MongoClient(uri);
//CONNECT TO MONGODB
var server = client.GetServer();
var db = server.GetDatabase(dbname);
var collection = db.GetCollection<BsonDocument>("systemdecorators");

//PARSE MONGO QUERY WITH USERNAME
var entityQuery = Query.And(
                    Query.EQ("username", PlayerPrefs.GetString("unme"))
                );
var find = collection.FindOne(entityQuery);
string ar = find.ToString();
ar = ar.Replace(",", "");
ar = ar.Replace('"', ' ');
//SEARCH COLLECTION FOR USERNAME REMOVING COMMAS TURNING INTO ARRAY
string[] words = ar.Split(' ');
uname.text = uname.text + " " + words[27];
fname.text = fname.text + " " + words[13];
lname.text = lname.text + " " + words[20];
age.text = age.text + " " + words[63];
eaddress.text = eaddress.text + " " + words[70];
mr.text = mr.text + " " + words[82];
```

The code above show how the profiles draw information from the mongoDB and displays it

C.3 AR Camera



Conceptual

Current Version

Fig 1 and Fig 2 shows the comparisons between the designs of our AR camera. This demonstrates the use of the royalty free icons used in the app (fig2)

```
//ADDED THIS FUNCTION.
public void delete(){
    print (_controller.lastActiveModel);
    foreach (var model in _controller.ActiveModels) {
        //Through all the models, if the current model is the one that was last selected, destroy it
        if (model.gameObject == _controller.lastActiveModel){
            Destroy(model);
            break;
        }
    }

    if (_controller.ActiveModels.Contains (_controller.lastActiveModel)) {
        _controller.ActiveModels.Remove (_controller.lastActiveModel);
    }
}
```

Fig.1
Controls the deletion of AR objects on the screen if you wanted to remove an object off the screen, you can simply click and delete it as a result of the method above.

C.4 Payment System


```
<html>
<head>
    <script language="javascript" type="text/javascript">alert("Accessing Interiar Secure Online Payment System, Click 'ok' to proceed")</script></head><font face="TimesNewRom
    </font>
<title>
Interiar
</title>
</head>
<body background="https://i.imgur.com/w9IZe7U.jpg">
<marquee width="100%"><font color="#FF0000"><b>Interiar Secure Payment System</b></font></marquee>

<form action="https://www.paypal.com/cgi-bin/webscr" method="post" target="_top">
<input type="hidden" name="cmd" value="_s-xclick">
<input type="hidden" name="hosted_button_id" value="URLHX7AAFFSQC">
<table>
<tr><td><input type="hidden" name="on0" value="Pay Designer">Pay Designer</td></tr><tr><td><select name="os0">
    <option value="Mark">Mark £175.00 GBP</option>
</select> </td></tr>
<tr><td><input type="hidden" name="on1" value="Enter your username here">Enter your username here:</td></tr><tr><td><input type="text" name="os1" maxlength="200"></td></tr>
</table>
<input type="hidden" name="currency_code" value="GBP">
<input type="image" src="https://www.paypalobjects.com/en_GB/i/btn/btn_paynowCC_LG.gif" border="0" name="submit" alt="PayPal - The safer, easier way to pay online!">
<img alt="" border="0" src="https://www.paypalobjects.com/en_GB/i/scr/pixel.gif" width="1" height="1">
</form>



</body>

<html>
```

This is the code for our payment system in html. This works perfectly fine in the browser and links to the PayPal payment system, which can send money to the correct user.

10.4 Appendix D: Quality Assurance

D.1 Functional testing

Test no.	Test Scenario	Example test case	Precondition	Test steps	Example test data	Expected outcome	Actual Results	Pass/Fail
1	Test if error message displays when an incorrect username or password is inputted within login screen	Test case 1: invalid username and valid password Test case 2: valid username and invalid password Test case 3: invalid username and password	Requires an Internet connection that doesn't have any restrictions (e.g. home network)	1. go to login screen 2. input data within username and password fields 3. press log in button 4. see if an error message is displayed	Test case 1: Username: "jzard420" password: "hello123" Test case 2: Username: martin007 password: "given456" Test case 3: Username: "nelloih43" password: "seven321"	should display an error message when user inputs a wrongs username or/and password	Does not display anything	Fail
2	Within create account, test if error message displays when user inputs a username that has already been taken	Test case 1: Input a username that is already taken	1. Requires an Internet connection that doesn't have any restrictions (e.g. home network) 2. must go to the create account screen	1. go to the create account screen 2. input an already taken username 3. see if it displays an error message	Test case 1: Username: "jmartinez"	Should display an error message when user inputs an already taken username	An error message displays	Pass
3	Within create account, test if error message displays when user inputs an email address in the wrong format (correct format – Text@emailProvider.com)	Test case 1: Input an email address in the wrong format	1. Requires an Internet connection that doesn't have any restrictions (e.g. home network) 2. must go to the create account screen	1. go to the create account screen 2. input an email address in the wrong format 3. see if it displays an error message	Test case 1: Email Address: "unknown@gt"	Should display an error message when user inputs the email address in the example test data	An error message displays	Pass

4	Within create account, test if the button works. It should display all the information user inputted within a newly created profile (e.g. first name: Martin, last name: Stein)	Test case 1: Input all fields in create account and see if it displays in profile screen	1. Requires an Internet connection that doesn't have any restrictions (e.g. home network) 2. must go to the create account screen	1. go to the create account screen 2. input all fields correctly 3. click create account button 4. see if profile displays correct information about user	Test case 1: Title: Mr First name: John Last name: Smith Username: jsmith007 Age: 40 Email: jsmith007@gmail.com	Should display a user profile with the fields mentioned in the test example data	Displays all the fields mentioned in the test example date within a newly create user profile	Pass
5	Within the profile page, test if the camera button opens the AR camera	Test case 1: Click the camera button	1. Requires an Internet connection that doesn't have any restrictions (e.g. home network) 2. must have created an account 3. requires you to log in to your account and enter your profile page	1. log in or create a new account 2. enter your profile page 3. click the camera button 4. see if it opens the AR camera	No data tested	Should display the AR camera after clicking the camera button	Opens the AR camera	Pass
6	Within the edit profile page, test whether the update profile button can update the user profile	Test case 1: Old information on profile screen about the user Test case 2: Updated information on profile screen about the user	1. Requires an Internet connection that doesn't have any restrictions (e.g. home network) 2. must have created an account 3. requires you to log in to your account, enter your profile page and then access the edit profile screen	1. log in or create a new account 2. enter your profile page 3. click the edit profile button 4. see if profile updates after you have entered all the fields correctly and have clicked the update profile button	Test case 1: Title: Mr First name: John Last name: Smith Username: jsmith007 Age: 40 Email: jsmith007@gmail.com Test case 2: Title: Mr First name: Johnson Last name: Smith Username: js9056 Age: 42 Email: JS9056@gmail.com	Should display a user profile with the fields mentioned in test case 2 example data	Displays the correct updated fields within the profile	Pass

7	Test basic usability of the application. i.e. whether it is possible to navigate to different screens	Test case 1: Login screen – navigates to profile page and create account screen Test case 2: create account - navigates to login screen and profile page Test case 3: profile page - navigates to login screen, edit profile screen and the AR camera Test case 4: Edit profile page – navigates to the profile page	1. Requires an Internet connection that doesn't have any restrictions (e.g. home network) 2. must have created an account	1. enter each of the mentioned screens mentioned in the test case. See if it navigates correctly	No data tested	Should navigate to the screens specified in the test case column	Navigates to the correct screens	Pass

D.2 Non-functional testing

Test no.	Test type	Test scenario	Pre-condition	Test steps	Expected outcome	Actual Results	Pass/Fail
1	Database Security testing	Test if mongo database displays hashed version of password under the password and confirm password field	Requires an Internet connection that doesn't have any restrictions (e.g. home network)	1. visit mlab.com and login to account 2. open the mongo database used to store user accounts 3. see if passwords have been securely hashed	All accounts should have a hashed password of length 29	All accounts successfully have hashed passwords of length 29	Pass
2	Usability testing	Test if all screens in the application are user-friendly	1. Requires an Internet connection that doesn't have any restrictions (e.g. home network) 2. must have created an account	1. open application 2. visit every screen (e.g. from the login screen to the create account screen to the profile screen etc.)	All usable features in each screen should be easy to use, stress-free and be simple for users to understand what usable features do what	After testing the application, we realised some of our screens were not user-friendly. For example, the create account screen had way too many input fields requiring user input. Another example would be the AR camera. There were buttons present in the screen that didn't have clear icons to represent what function they carry out	Fail

3	Database recovery testing	Test if there is a recovery plan if database gets corrupted or erased	Requires an Internet connection that doesn't have any restrictions (e.g. home network)	1. visit mlab.com and login to account 2. open the mongo database 3. go under backups 4. see if there is a mongodump or a way to create one	Their should be a recovery (mongodump) of the database containing all user accounts	There is no recovery (mongodump) present. Having a recovery requires payment with MLab. Paying for stuff is something we tried avoiding in the project	Fail
4	Compatibility testing	Test if AR camera has no compatibility issues when using on a mobile	1. Requires an Internet connection that doesn't have any restrictions (e.g. home network) 2. must have created an account	1. login to your account in the app 2. click the camera button in your profile page 3. see if it opens the AR camera without any rendering issues	The AR camera should open up	Most mobile we tested the app with seemed to open the AR camera, but their were issues with certain mobiles such as Samsung Galaxy S7 Edge. The camera screen would not either open or it would display a load error message	Fail
5	Database scalability testing	Test if our Mongo database allows us to create at least 100,000 user accounts before exceeding 500 MB (we would need to pay £15 or over to add more storage. Under 500 MB is free to use)	Requires an Internet connection that doesn't have any restrictions (e.g. home network)	1. visit mlab.com and login to account 2. open the mongo database 3. see how much storage each account roughly takes 4. calculate how many accounts can be created that doesn't exceed 500 MB	The database should be able to hold over 100,000 user accounts	The database can hold roughly 149,700 user accounts. Each account on average takes 3.34 kilobytes. Multiply this by 149,700 and it should be equal to 499998 kilobytes, equivalent to 499.98 megabytes	Pass
6	Load testing	Test if app can handle multiple users at once	1. Requires an Internet connection that doesn't have any restrictions (e.g. home network) 2. must have created an account	1. get around 10 to 15 people to login to their account at the same time 2. see if there are any loading issues	The application should work normally. There shouldn't be any technical issues	The application seems to be running smoothly when exactly 10 users logged in at the same time. The loading speed of the app accessing the user profile seemed normal. Loading speed for other screens also remained consistent	Pass

D.3 Static and dynamic testing

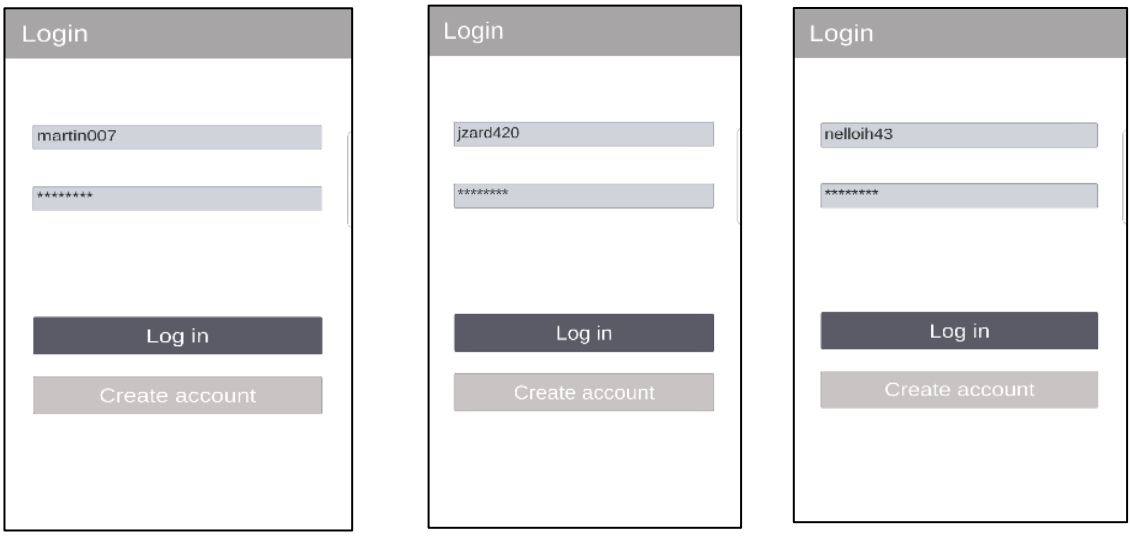
Test no.	Screen/s	Test type	Defects to search for	Tools used	Test steps	Expected outcome	Actual Results	Pass/Fail
1	Create account	Static testing	1. syntax violation 2. dead code 3. unused variables 4. variables with undefined value or values that are rendered useless 5. causes that might've made code too long or difficult to understand	Built-in Automated tools	1. open all relevant C# scripts 2. review each script and find any of the defects mentioned	To find at least one or more defects within each of the relevant C# scripts	**Multiple defects found!** 1. dead code 2. unused variables 3. variable equal to null that'll produce a NullPointerException. Thi s variable only produces null, if user inputs a username in the username input field that does not exist in the database. This variable essentially queries the database to find the inputted username. If it doesn't find the username. It returns null. This is a defect I cannot change	pass
2	Login Screen	Static testing	1. syntax violation 2. dead code 3. unused variables 4. variables with undefined value or values that are rendered useless 5. causes that might've made code too long or difficult to understand	Built-in Automated tools	1. open all relevant C# scripts 2. review each script and find any of the defects mentioned	To find at least one or more defects within each of the relevant C# scripts	**Defect found!** 1. variable equal to null that'll produce a NullPointerException. This variable only produces null, if user inputs a username & password that doesn't exist in our database i.e. invalid username or password. This is a defect I can't change	Pass
3	Profile Screens	Static testing	1. syntax violation 2. dead code 3. unused variables 4. variables with undefined value or values that are rendered useless 5. causes that might've made code too long or difficult to understand	Built-in Automated tools	1. open all relevant C# scripts 2. review each script and find any of the defects mentioned	To find at least one or more defects within each of the relevant C# scripts	No defects found!	Fail

4	Create account	Dynamic testing – unit testing	1. unexpected exceptions 2. program not being able to detect text being inputted in the input field each time 3. unexpected errors	Unity console	1. run the create account scene in Unity 2. enter text in all input fields (e.g. first name, last name etc.) 3. see if inputted text prints & updates in console	To see if the inputted text within each of the input fields is recorded by program. And to see if the program updates each time the input fields get updated with new text **Expected output:** first name: Gina last name: Begum username: gina008 email address: gina123@gmail.com password: hello123 confirm password: hello123 age: 30	Expected output matches actual output! No defects found	Fail
5	Login screen	Dynamic testing – unit testing	1. unexpected exceptions 2. The Mongo queries placed within C# Script not being able to find a valid username & password in the database 3. unexpected errors	Unity console	1. run the login scene in Unity 2. first enter a valid username & password (you will need to create an account) 3. then enter an invalid username or password 4. see what the console prints	When user inputs a valid username and password, it should output the whole document extracted from MongoDB in String format When user inputs a invalid username or password, it should output just NULL in console. **Expected output for valid username and password:** { "_id" : ObjectId("5aa1cde2fee0042de8bd96d6"), "firstname" : "Ishtiyaq", "lastname" : "Ali", "username" : "ish2nv", "password" : "0aa7a662c728b7407c54ae6bfd27d1", "cpassword" : "0aa7a662c728b7407c54ae6bfd27d1", "day" : null, "month" : null, "year" : null, "age" : "44", "eaddress" : "hello@gmail.com", "phonenumber" : null, "title" : "Mr", "account" : "Decorator", "companyname" : "Topps Tiles", "jobtitle" : "Tiler" **}** **Expected output for invalid username or password:** Null	Expected outputs matches actual outputs! No defects found!	Fail

6	Profile Screens	Dynamic testing – unit testing	1. unexpected exceptions 2. profile screen not displaying user details or displaying incorrect details about user 3. unexpected errors	Unity console	1. run the login screen or create account screen first 2. navigate your way through the program to the profile screen 3. see what is printed on console and on the profile screen. Is it correct?	All fields within the profile screen should have correct information **Expected output:** Title: Mr First name: Ishtiyaq Last name: Ali Username: ish2nv Age: 30 Email: ishtiyaq93@gmail.com	Expected outputs matches actual outputs! No defects found!	Fail
7	1. Create account 2. Login screen 3. Profile Screens 4. Edit profile screens 5. AR camera	Dynamic testing – integration testing	1. unexpected exceptions 2. any unexpected errors	Unity console	1. integrate all the scenes in one Unity file 2. see if any errors come about in the Unity console	All scenes should work when put together in one Unity file	All scenes work successfully together in a Unity file	Fail

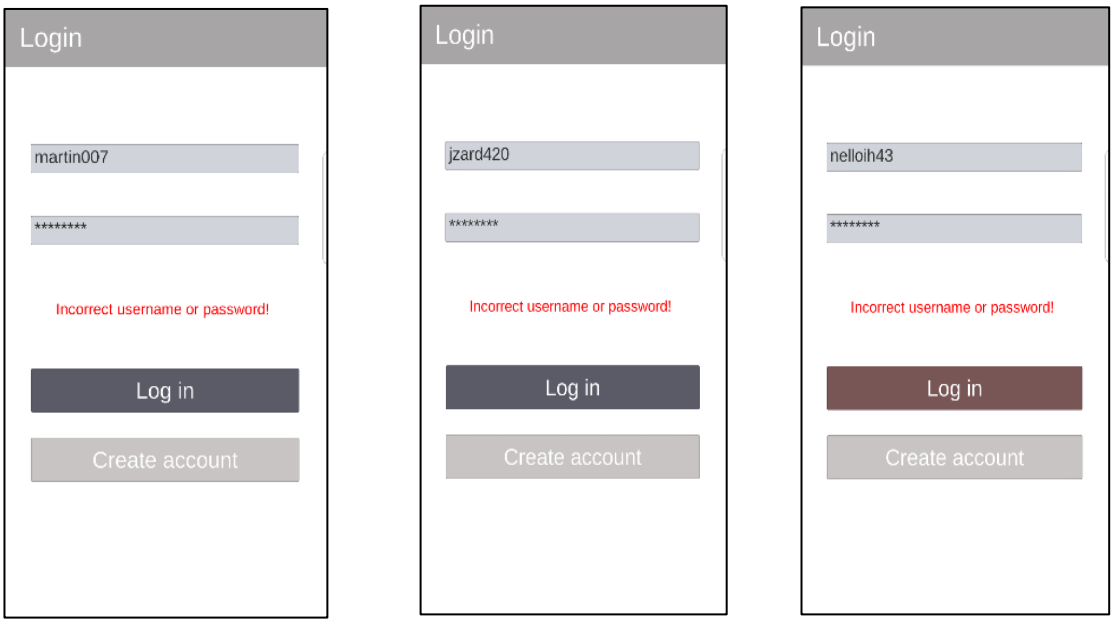
D4 Evidence and results – functional testing

Test 1



No error message displayed in any of the screenshots when entering incorrect username and/or password. This is a problem, since user will not know whether they are making a mistake or whether the app is not responding

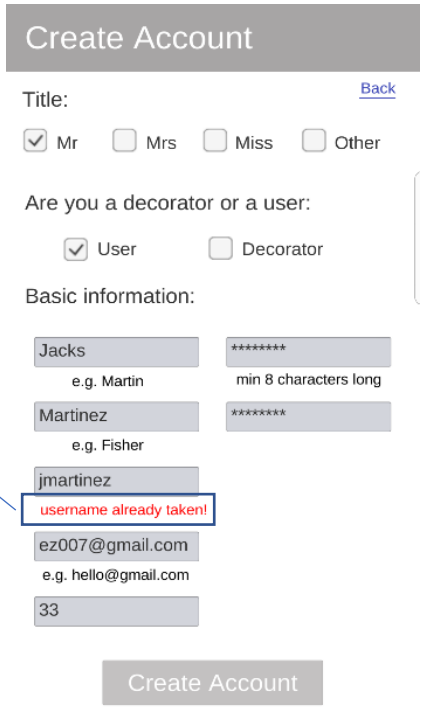
Changes to fix error



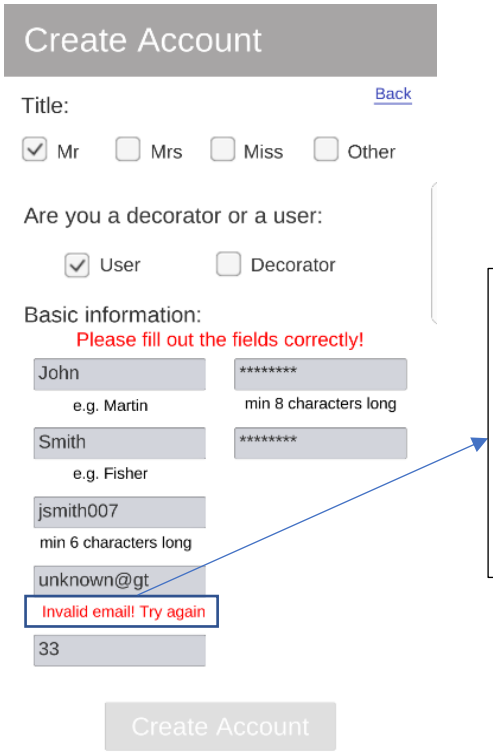
When the log in button is pressed it now displays an error message in red. The awareness from this functional test enabled me to notice this mistake and rightly fix it immediately

Test 2

The example data “jmartinez” produces the following error message “username already taken.” This is proof that the error message pops-up when user inputs a username that has already been created

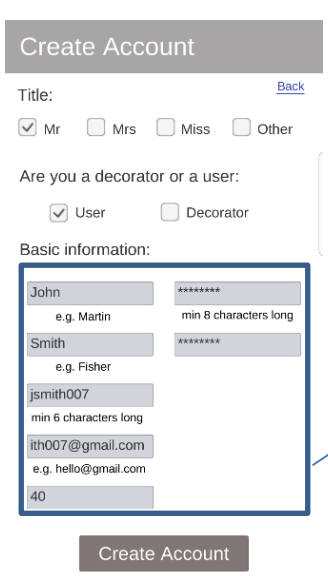


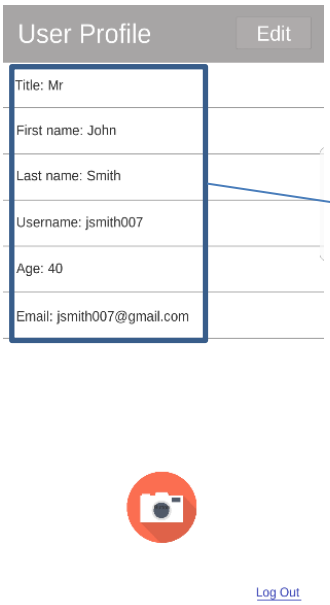
Test 3



The example data “unknown@gt” produces the following error message “invalid email! Try again.” This is proof that the error message pops-up when user inputs an email address in the wrong format. This is a way of validating the reliability of the email addresses users input

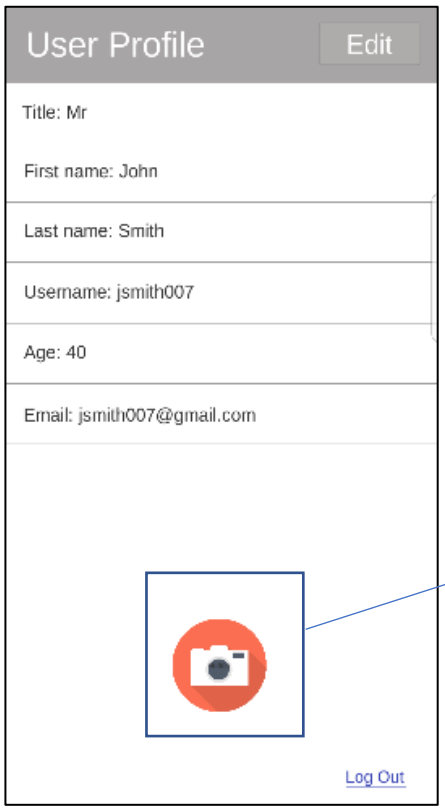
Test 4





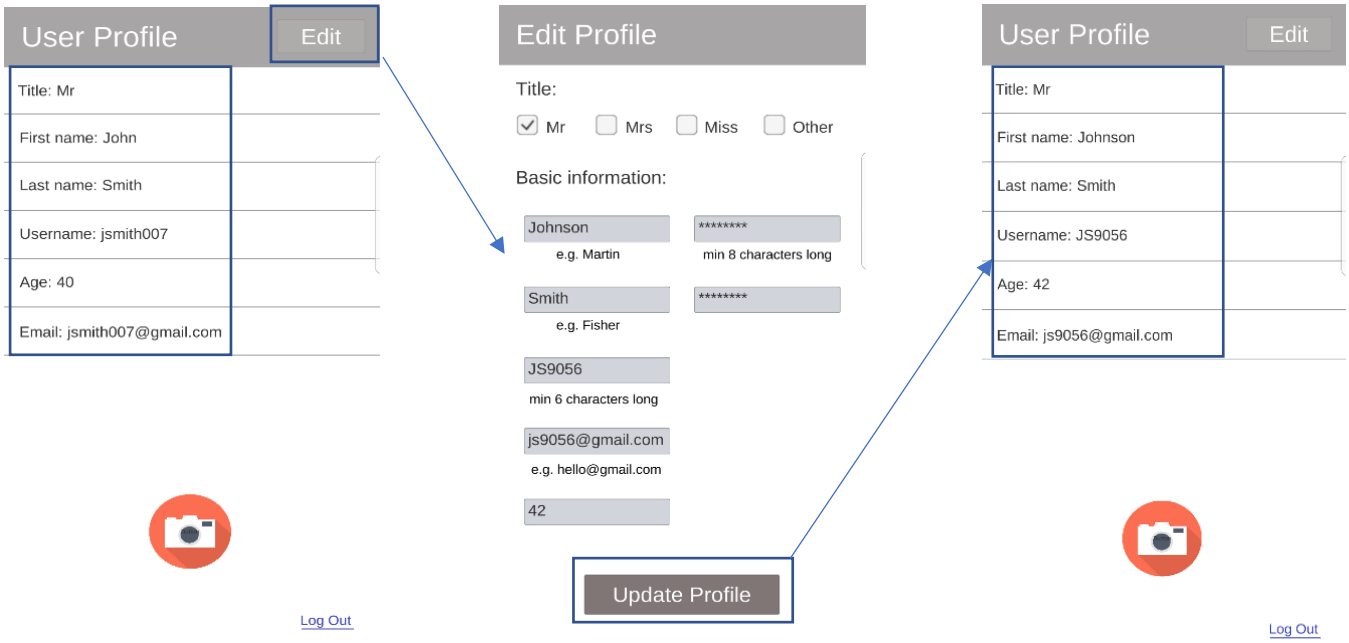
First, we create an account. Then enter our user profile. As you can see, the text inputted in the fields within the create account screen display within the user profile, meaning the user profile was able to extract information about the user correctly and display it

Test 5



Camera button within the profile page can successfully load the AR camera and display it on the mobile screen. The load time speed is usually between 1 to 5 seconds, which is not bad considering how technical/advanced an AR camera is

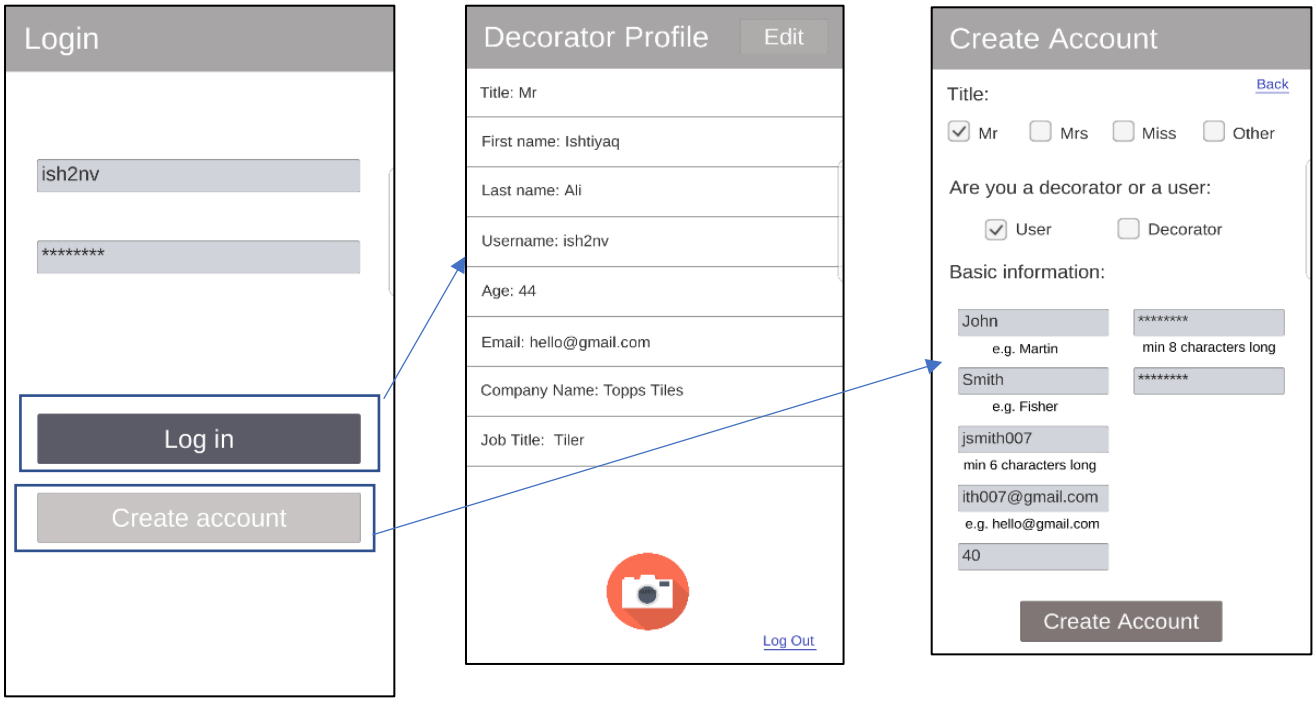
Test 6



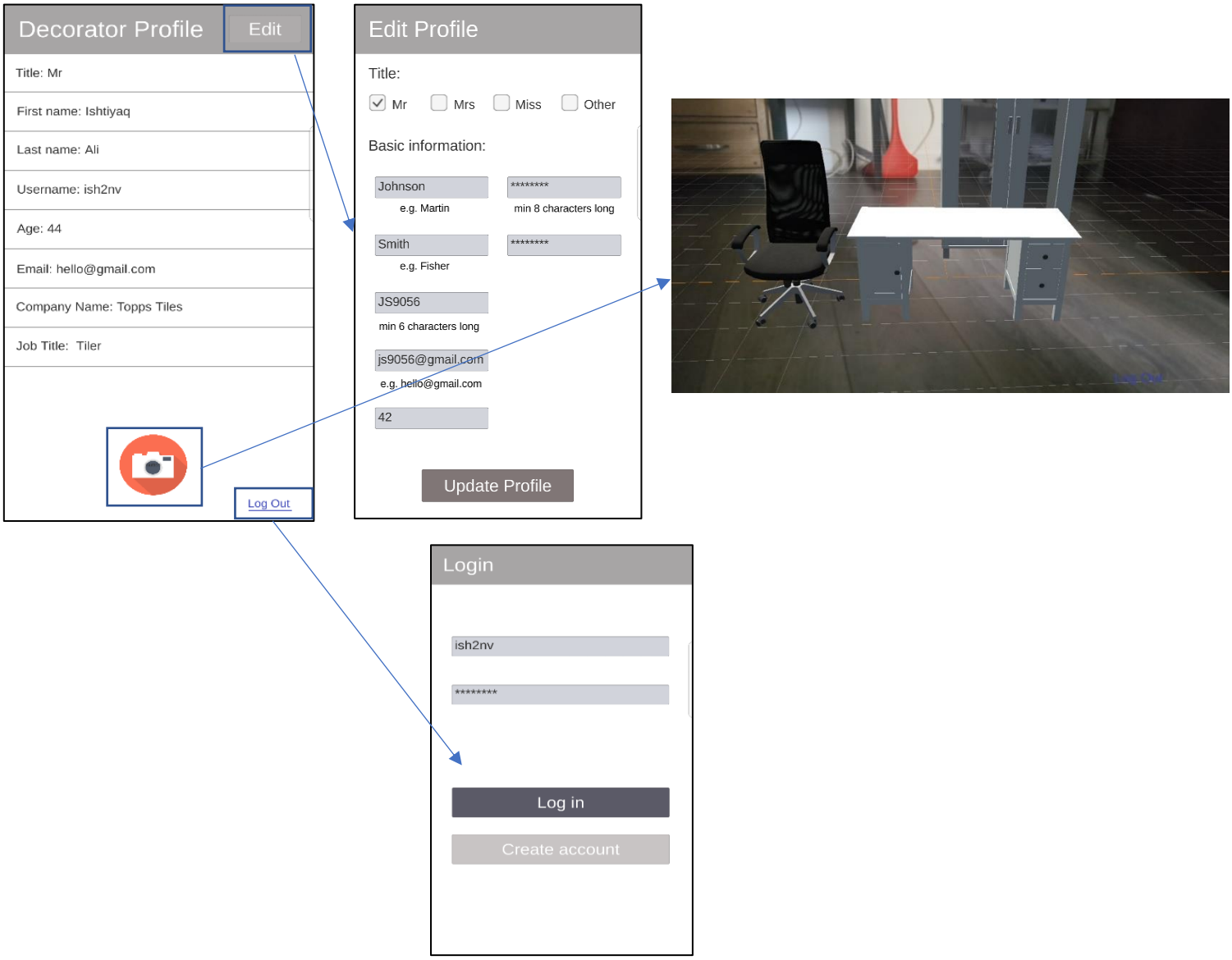
These series of screenshot are trying to illustrate how our application can update a user's profile. In the first screenshot it shows fields with user information. Second screenshot shows the user going into the edit profile screen. This is where the user updates their profile. Last screenshot shows the user profile displaying the updated user information.

Test 7

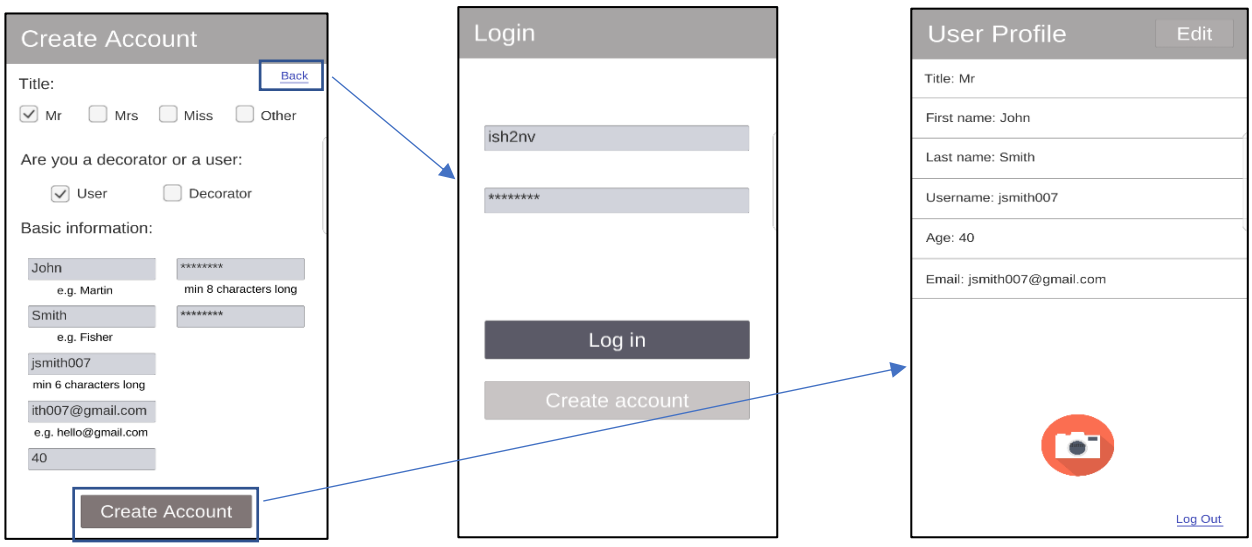
Login screen – navigates to profile page & create account



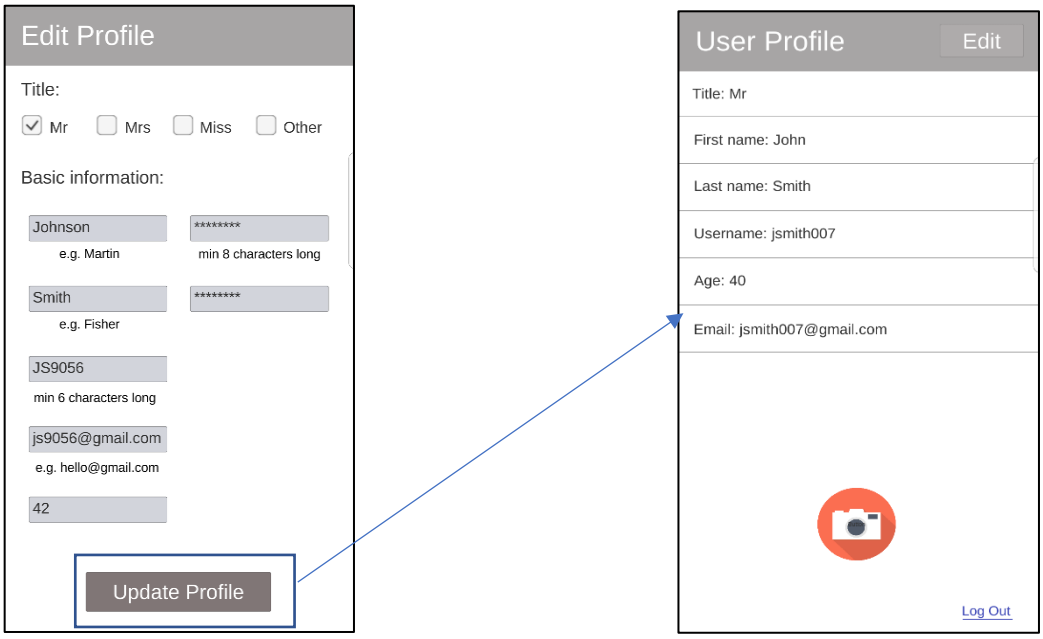
Profile page – navigates to edit profile screen, AR camera & login screen



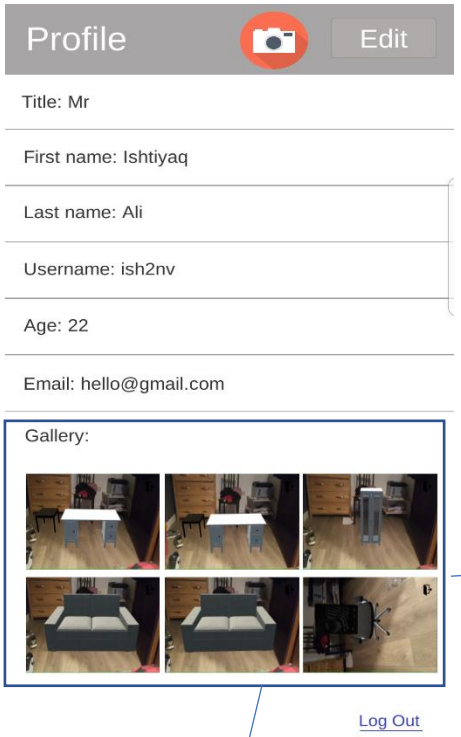
Create account screen – navigates to the log in screen and the profile page



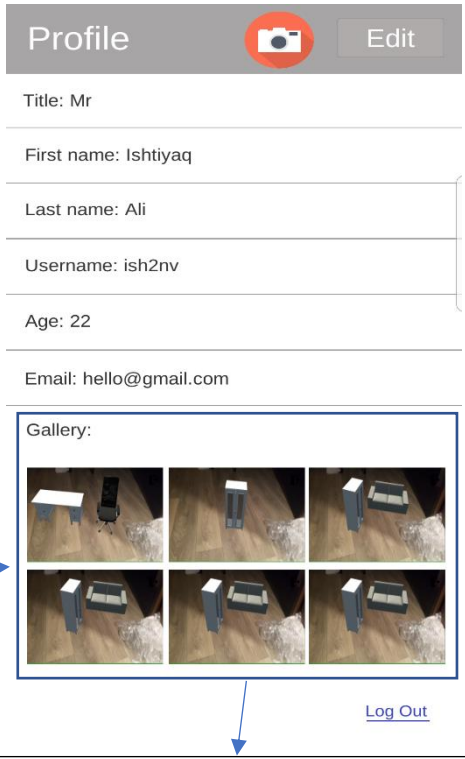
Edit profile page – navigate back to the profile page (with updated user information)



Test 8



All images do display in the gallery within all available slots



As you can see now images have changed. This is because all images have been overwritten when users take more screenshots. This is something that we didn't want our app to do, however it is something we couldn't figure out how to fix in time

D.5 Evidence and results – non-functional testing

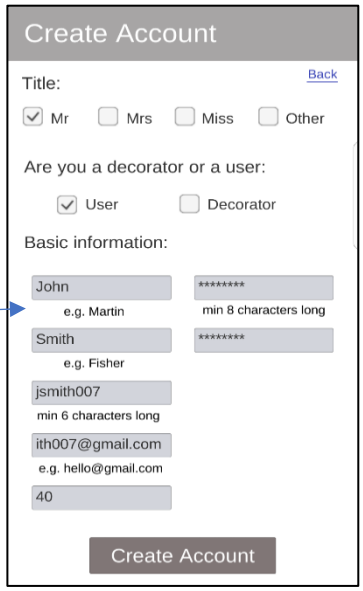
Test 1


```
{
  "_id": {
    "$oid": "5aaa700ef36d2876ecd3fa3a"
  },
  "firstname": "Johnson",
  "lastname": "Smith",
  "username": "JS9056",
  "password": "0aa7a662c728b7407c54ae6bfd27d1",
  "cpassword": "0aa7a662c728b7407c54ae6bfd27d1",
  "age": "42",
  "eaddress": "js9056@gmail.com",
  "title": "Mr",
  "account": "User"
}
```


Within our Mongo database, accounts are represented as documents. Within this document we can see proof that the password and confirm password fields have indeed been hashed. This adds a layer of security to our database since hackers will not be able to decrypt the password nor will developers have access to it. Thus, securing users privacy

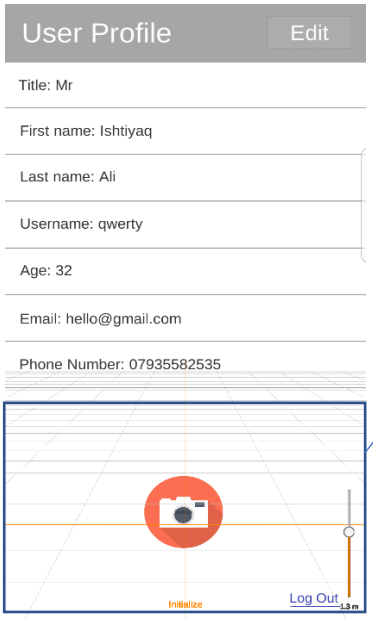
Test 2

Changed the create account layout to…



Due to how congested the create account looks with multiple input fields squashed together, we decided to reduce the number of input fields, so it is more user-friendly i.e. preventing users from inputting too much information

Test 4



The AR camera in our application opens almost instantly. Loading speed is at a very good rate for most devices. But there are certain devices that have issues in opening the AR camera. Some may not open on the first or second try. And some do not open the AR camera at all. The screenshot on the left was made by a Samsung Galaxy S7 Edge. As you can see, the AR camera has not been rendered properly. This is an issue we did solve for Samsung phones, but there are other mobile devices that we know cannot render/load the AR camera properly.

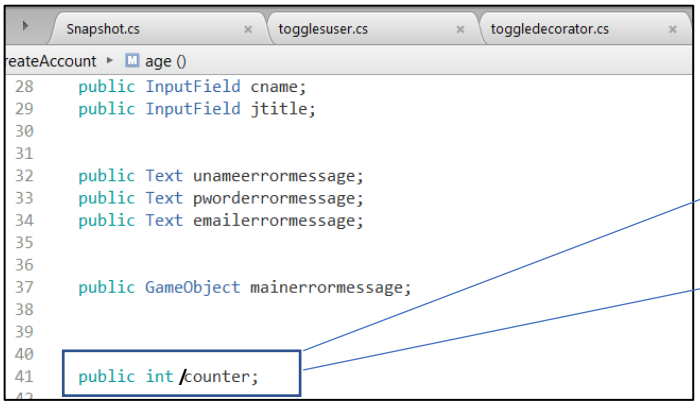
Test 6

	User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10
Loading speed of user profile	1.1	1	0.8	0.9	0.9	1.5	1.7	1.3	1.8	1.1
Loading speed of AR camera	1.9	2.3	2.1	2.4	2.7	2.1	2.3	2.5	3.4	3
Loading speed of create account screen	1	1.5	1.7	1.8	1.2	1.4	1.2	1.9	0.8	1.2
Loading speed of edit profile screen	1.6	2.3	2.5	2.9	2.6	2.3	2.5	2.1	2.2	2.3

The ten users we had, whom all logged in at the same time all recorded the time it took (in seconds) to load each screen within the application. From this table I was able to conclude that there were no issues in loading any of the screens. All the loading times seem reasonable.

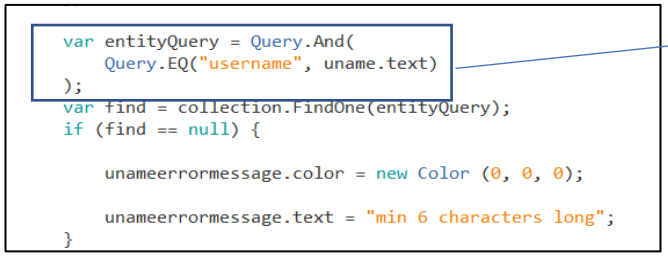
D.6 Evidence and results – static and dynamic testing

Test 1

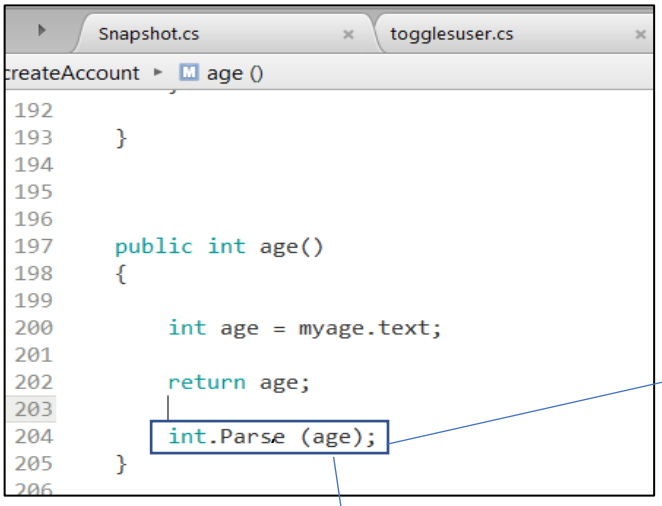


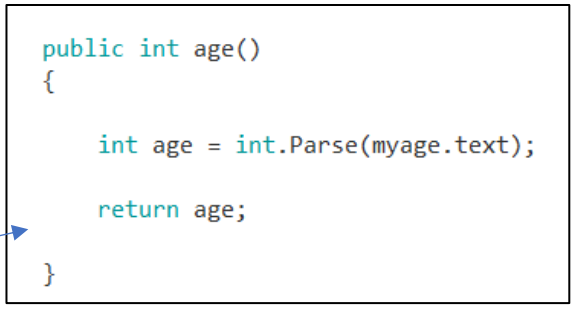
Unused/uninitialized variable

Variable removed



Returns null if it cannot query the inputted username. This generates a NullPointerException. This is a defect I cannot remove since I need this query to search for the inputted username in the database and there is no other way of carrying this out





Correctly changed

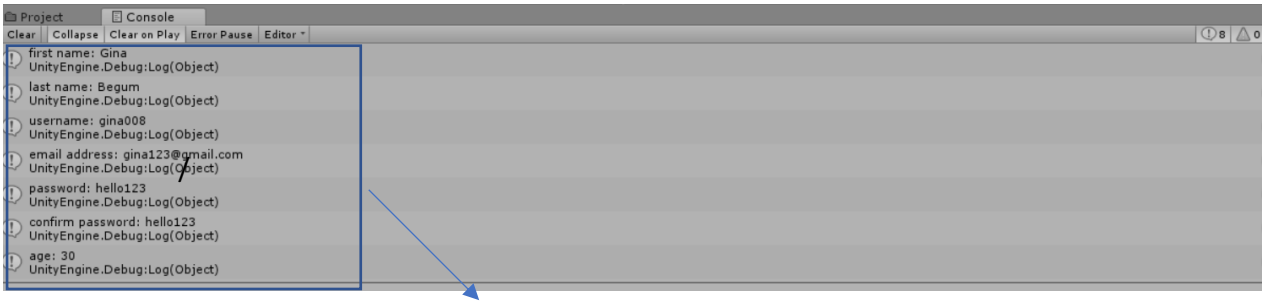
Dead code

Test 2

```
var entityQuery = Query.And (
                        Query.EQ ("username", a.username),
                        Query.EQ ("password", a.password)
                    );
var find = collection.FindOne (entityQuery);
Debug.Log (find);

if (find == null) {
    errormessage2.SetActive (true);
} else {

    errormessage2.SetActive (false);

}
```

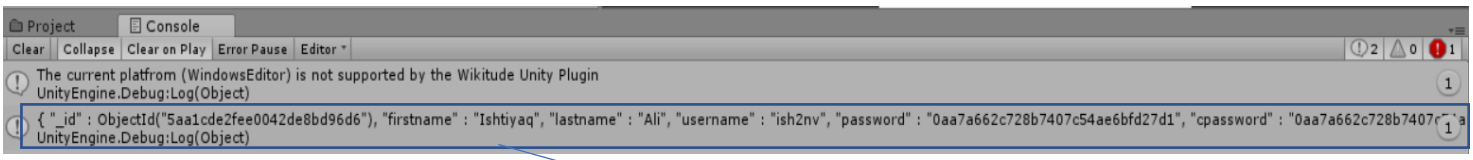
entityQuery is a variable that produces a NullPointerException. This variable only equals null, when user inputs a username or password that doesn't exist in our database. This is a defect I can't change since this is the only way in carrying out this important task. I've tried the try and catch block to override the default behaviour of the program, but this didn't really help

Test 5



Proof that this test was successful, since the expected output has been printed into our Unity console

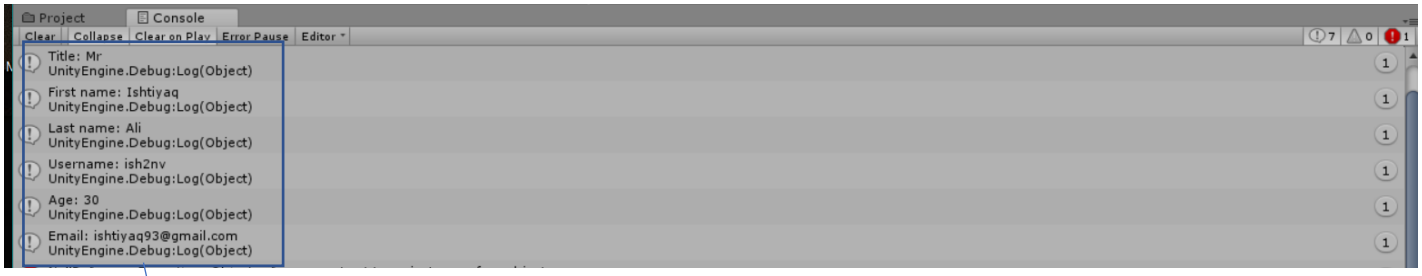
Test 6

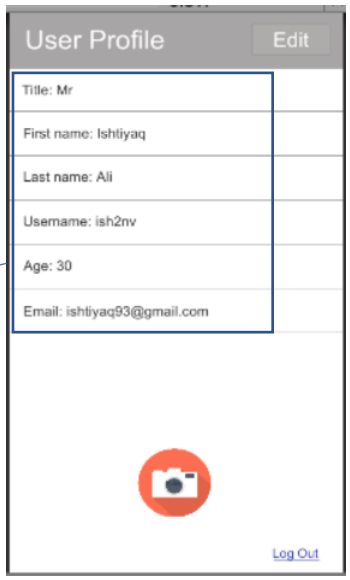


Unity console is printing out the MongoDB document that was found from inputting a valid username & password in the login screen

Unity console printing out null, since an invalid username & password was inputted in the login screen

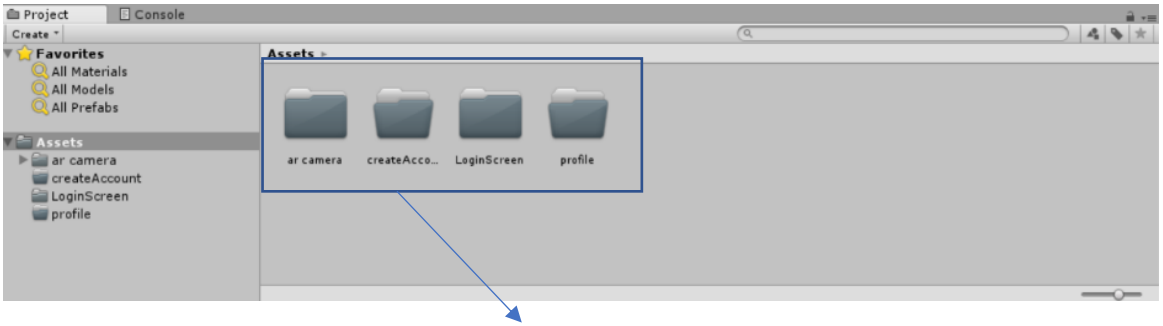
Test 7





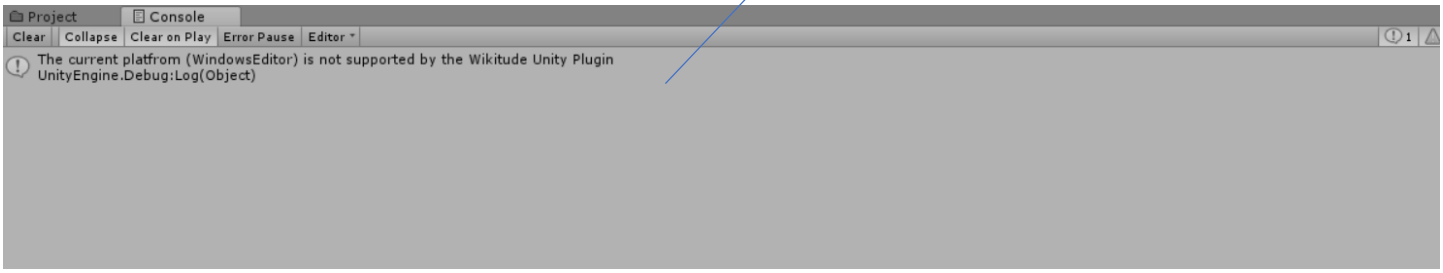
Expected output has been printed in the Unity console and on the profile page. These screenshots are proof that this test was successful

Test 9



All unit tests have been integrated together in one Unity file

Proof that the integration was successful as there are no error messages. The Unity file can also be built to Android without any issues



10.5 Appendix E: Summative Evaluation

E.1 User Manual

1. You will be prompted to either log into an existing, or create a new account and fill out the relevant details. When creating an account, please ensure that you follow all the onscreen prompts regarding your username, password etc.
2. If this is a new account you're creating, you will be prompted to log back in using your credentials. If this is not a new account, skip to step 3.
3. You will then arrive at the profile view, where you have the option to view your details, edit your profile or press the camera icon in the top panel which will take you to the live AR camera screen.
4. Once in the camera screen, drag the slider on the right to roughly match the height of your device from the ground, this helps with perspective.
5. After the height is selected, face the camera towards the area you wish to augment and initialise the process by pressing the button in the bottom middle of the screen.
6. Now you have the option to select furniture by pressing the button in the top left, choosing a category and then dragging the furniture from the menu and dropping it into the live view.
7. To manipulate the furniture in the augmented view, a user can:
    - Drag an object around using a single finger.
    - Scale an object using a pinching motion with two fingers.
    - Rotate an object using a twisting motion with three fingers.
    - Delete an object by tapping the object and then pressing the dustbin icon.
8. After adding any desired furniture and positioning accordingly, you can take a snapshot of the augmented view which will save to their profile. This is done by pressing the circular lens icon on the right hand side.
9. If you wish to change the view to a different place, press the reinitialise button which takes you back to step 3.
10. To exit out of the camera view entirely, and go back to the user profile; simply press the door icon in the top right. Here you can see any snapshots taken in your personal gallery on your profile.

E.2 Link to code repository

http://gitlab.doc.gold.ac.uk/enewe001/interiAR-group-H-code