Online Supermarket System

*Emre Selcuk*

*Basar Aksanli*

*C. Haluk Sumen*

*Kadir Ozcan*

*Daniel Holmes*

**Contents:**

1. **Introduction**
2. **Initial Need Finding**
3. **Conclusion of Needs**
4. **Initial Storyboards**
5. **Interview**
6. **Iterations**
7. **Testing**
8. **Final Application**
9. **Conclusion**
10. **Introduction**

Introducing our idea for application, we decided we would like to create an application allowing for the easy shopping of a supermarket, which is delivered to our home quickly and easily, similar to Deliveroo for food. We believe that this application will prove to be very useful and a viable product to be used on the market, which would require a good interface in order to attract a lot of users, and to make it easier for orders to transact. To follow this, we carry out need finding.

1. **Initial Need Finding**

For the initial need, finding process we decided it was a good idea to begin with gaining a general idea of the market to see which users would be interested in our product and which precise needs they require being satisfied. First, we carried out a survey.

***Initial Survey***

In the first step of gathering needs to conclude, the needs we created a questionnaire which we wanted to send out to 50 people. However, as we concluded, we had open-ended questions so we decided it might be better to carry out 40 questionnaires sent to anonymous users, and then 10 people we carried out pager studies and we asked them in person their opinions.

The questions were as follows:

**What is your age group?**

* 80%: 18-25
* 20% 26-40

**What is your Gender?**

* 60%: Female
* 40%: Male

**Do you frequently shop at supermarkets?**

* 80%: Yes
* 20%: No

**Do you often need to travel far to get the groceries you need?**

* 80%: No
* 20%: Yes

**Would you be interested in having groceries delivered to your doorstop similar to Deliveroo?**

* 80%: Yes
* 20%: No

**Have you heard of any apps such as Glovo or services that deliver groceries?**

* 60%: No
* 40%: Yes

**Have you ever used a service to deliver groceries?**

* 60%: No
* 40%: Yes

**If yes, what did you like about the service? Alternatively, what did you dislike?**

*Few responses to this open questions, this is a summary.*

* *Finding a lot of options*
* *Doorstep delivery*
* *All products in perfect condition*
* *Unavailable items changed to a close substitute*
* *Good prices*
* *Dislike big fee for services.*

**If no, what would encourage you to use such a service?**

*Few responses to this open question, this is a summary*

* *The quality of products*
* *Reliability*
* *Low costs of service*
* *Can’t find what want in local supermarket*
* *Living far away from shops*
* *No service fee*
* *Quick delivery*

**What features would you like to see on an app such as this?**

*Few responses to this open question, here is a summary:*

* *Suggestions on what to buy based on season*
* *Point out where the producers are*
* *Names on the products, such as Lush branding*
* *Section to order organic and zero km production*
* *Paying with Paypal*
* *Easy Handling*
* *App for free*
* *Selection of hour of delivery*
* *Various lists of products that they are buying, such as for a party, football, Christmas dinner etc.*
* *Some items they can buy before final checkout based on recommendations*
* *VR shopping*
* *Different products*
* *Customize offer to user needs, such as what they like etc.*

After processing all of the relevant data collected, we concluded that people would be interested in our application; however, the way in which the app flow is to be developed is an absolute must. In addition, it needs to be extremely easy to use and allow for delivery almost quickly, and with a low or no fee.

The features people were interested in seemed interesting at first, however the most common features which were requested where the following:

* Easy Handling
* App for Free
* Different Products

Therefore, we will aim to make this app as easy as possible to use, and of course, the app will pay for the products they buy but the app itself will be free. As for the different products, this will be the main selling feature of our app in that we will offer as many product categories as possible.

1. **Conclusion of Needs**

From the initial survey, we found that there is in fact a need for an app, which allows for the easy shopping of a supermarket, which is online. One need we did not anticipate is the need to show as many different products as possible. We found that user are in fact interested in such an app as long as it is easy to use and easy to order food, and it offers wide selection of products. Therefore, the following needs need to be satisfied:

* Easy to use/Easy handling
* App is free
* Different products

**Easy to use/Easy Handling**

This means we will need to make the interactions between the interface and the user as easy as possible, in order to allow the user to quickly order what they need. This is also accompanied by the easy handling of products, so the user wants to know for sure that the products they are ordering will be easily delivered to them, and the order is easy to handle.

For this, we will take payment via cash at the door to reduce difficulties for the user, and we will make the interface flow as intuitive as possible.

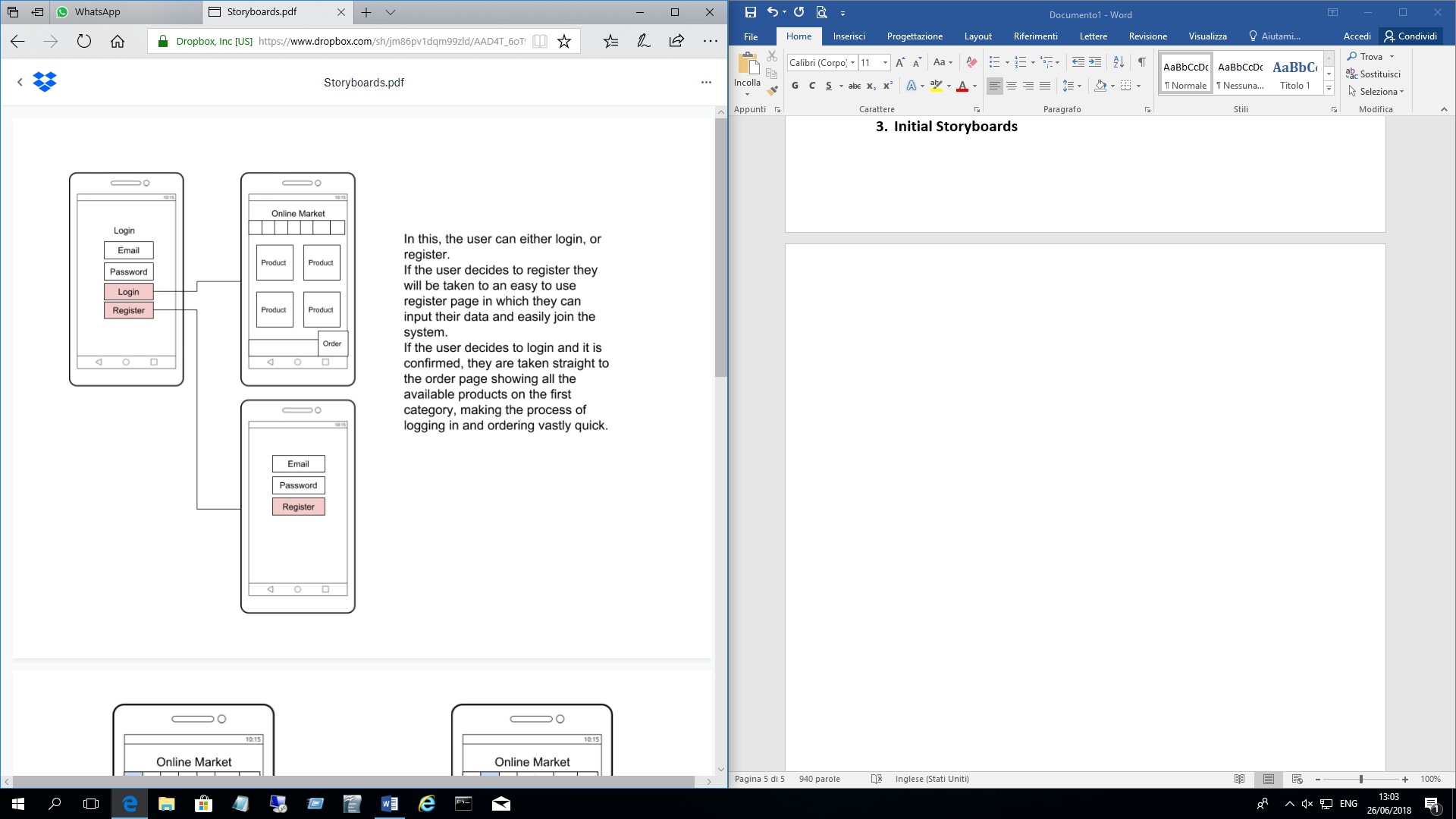
**App is free**

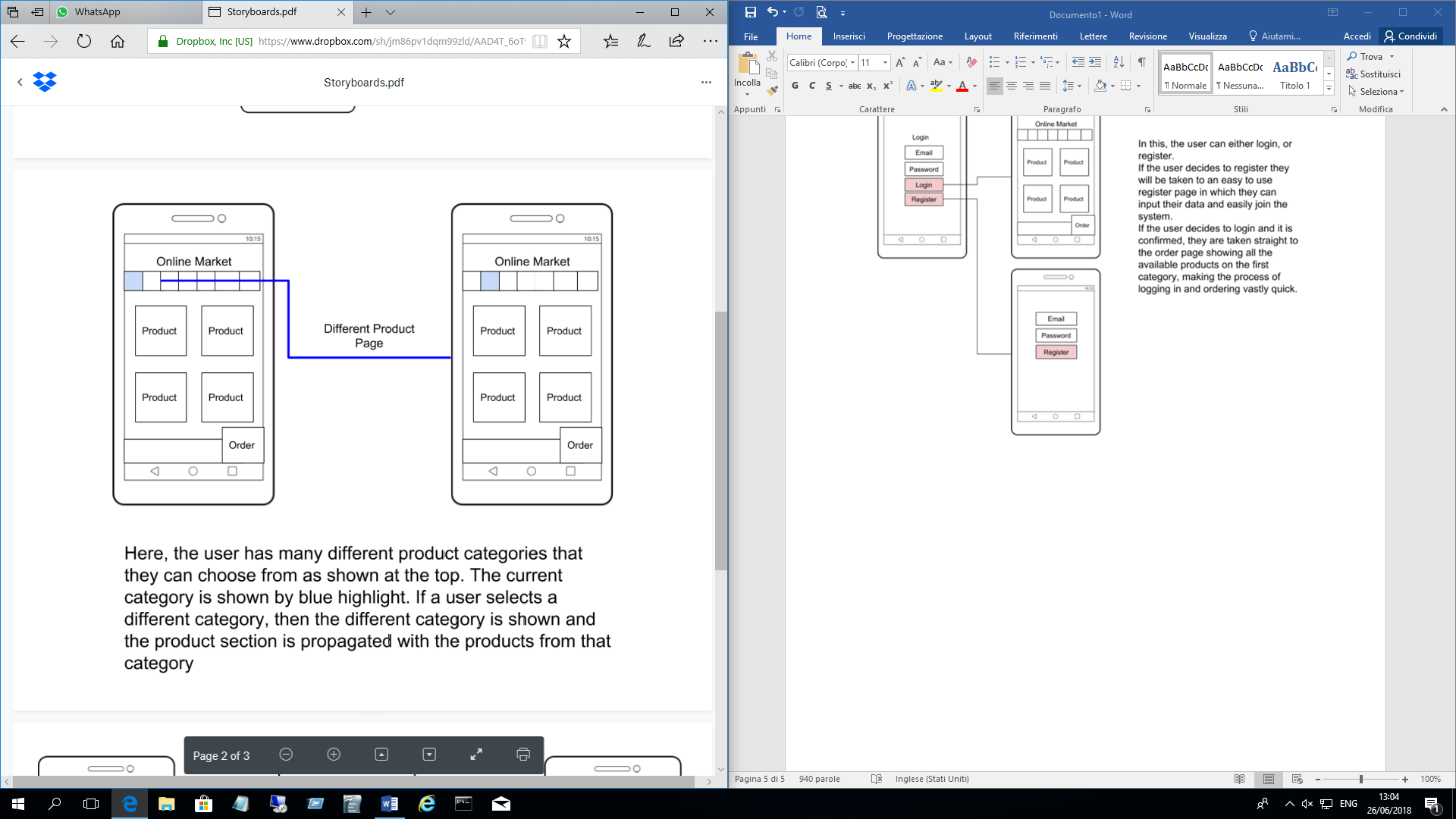
This is self-explanatory, of course, the user will pay for the products that they order, but the very app itself will be free on the app store.

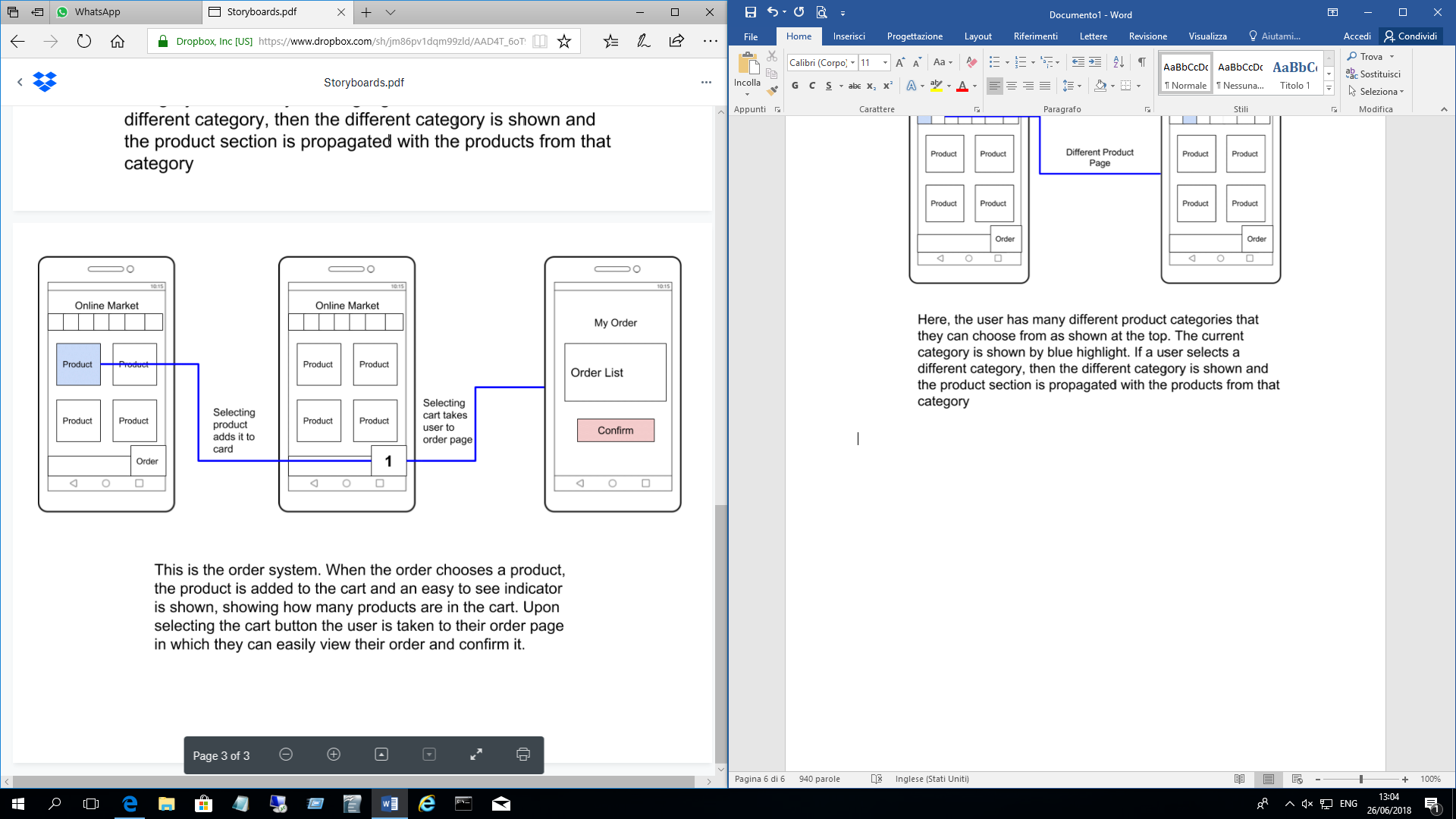
**Different Products**

Many users wanted to be able to have a wide selection of different products, to which they can browse and select what they want. In addition, they want these products to be categorized. Therefore, to do this we will develop separate areas for each category and allow the user to select which category they desire in order to allow for a wide range of products.

1. **Initial Storyboards**







We developed the following storyboards to satisfy the needs we identified from users from throughout initial survey.

1. **Interview**

For an initial interview, we carried out paper prototypes, which we developed in comparison of our storyboards, so the paper prototypes simple reflected those above. We decided to aim for 20 people to interview, and we showed them our initial paper prototypes. Unfortunately, we did not record the entirety of the interview, but we recorded the answers to some set questions, which we asked about the product, follows is the results:

**Did you like the application?**

* 12 Yes
* 8 No

**If yes, what did you like?**

* Simplicity
* Easy to use
* Handling of order

**If no, what did you dislike?**

* App seems ‘clunky’ i.e. a lot of information in one place
* Did not see the point of it as they could easily shop themselves.

**Could you see yourself using this in the future?**

* 8 Yes
* 7 No
* 5 Don’t Know

After these initial questions, we adapted our questions based on the responses of the users, if they said no to like it, we asked those people different questions from the people who said yes.

**To the people who said “no”:**

1. **What would you change?**

* Nothing, they disliked everything
* More Categories
* More Companies
* More zone support

1. **Do you prefer to shop in person?**

* 5 Yes
* 3 Don’t Mind

1. **Have you heard of Glovo?**

* 2 Yes
* 6 No

1. **Do you like Glovo? What do you like?**

* 6 Never heard of it
* Like the easiness to use

1. **Do you use any other applications like this?**

* 7 Said Mo
* 1 Said Yes

1. **If yes, what is different from them to ours?**

* More products to choose from

**To the people who said “yes”:**

1. **What would you change?**

* More categories
* More zone support
* Customer Service
* Credit Card
* Compensation for damaged products
* Paper Receipt

1. **What would you keep?**

* Interface to choose products
* Category selection

1. **Do you like the idea of shopping at home?**

* 11 Yes
* 1 No

1. **How much would you be willing to pay for delivery?**

* Free
* Less than 5 euros

1. **How frequently can you see yourself using this application?**

* 10 = Once a week
* 2 = Everyday

From what we gathered we found that the idea of the project was simple, and a genuine need that users require to be satisfied, as eight people they would use it. From this general interview, we gained a clear idea in the direction we must drive out product. In its simplicity and easy to use factor for shopping.

1. **Iterations**

**Sprints:**

We decided that a suitable sprint time would be 2 weeks; we decided during the planning process that we would decide on a set of features to develop during the sprint, and then develop it. To follow we would carry out user research.

**User Research:**

For the user research we decided that we want to talk to at least 10 people at the end of each iteration, and then carry out a final survey on the last iteration which actual customers, and cashiers at the supermarkets.

**Iteration 1**

From the need finding phase, we found that the need for a good product page was required. Therefore, for the end of this sprint, we wanted to develop the general ability to view products of a category, and this required us to be able to see many different products at once within a page. This meant setting up the required database and then adding products to the database from a selected supermarket.

Firstly, we sent an email to Carrefour, requesting if we could create the application, and if possible have a list of products, they would like us to add to the application. Unfortunately, they did not reply so we decided to send someone in who then individually wrote down all of the *common* products that were sold by Carrefour. Because if a product is uncommon we cannot ensure the ability to get this product every second of every day, perhaps in the future we could add a section to add and delete products.

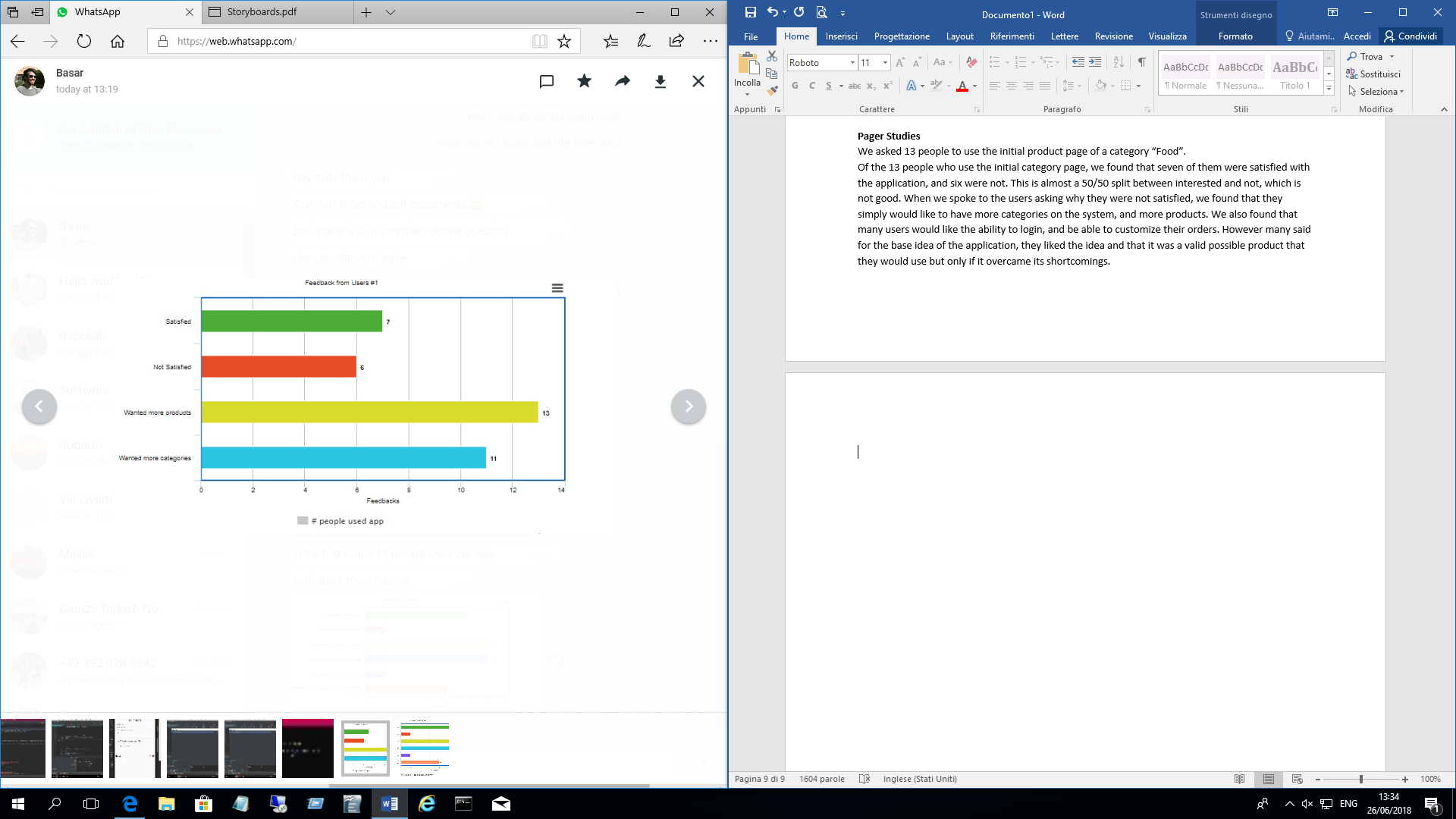
Secondly began the programming of the pages and creating the link to the now populated database of products, which would include at this point, 20 products. We finished just this page and carried out the survey with 10 people on the street, carrying out a simple pager study.

We did this across one day during 1 PM directly after our first sprint, following is our results:

**Pager Studies**

We asked 13 people to use the initial product page of a category “Food”.

Of the 13 people who use the initial category page, we found that seven of them were satisfied with the application, and six were not. This is almost a 50/50 split between interested and not, which is not good. When we spoke to the users asking why they were not satisfied, we found that they simply would like to have more categories on the system, and more products. However many said for the base idea of the application, they liked the idea and that it was a valid possible product that they would use but only if it overcame its shortcomings.



**Conclusion of study**

To conclude this study we found that the users generally like the idea of this system, and would likely use it in the future if it were to overcome its shortcomings. However, they would like more products and categories to be on the system. Therefore, for this sprint, we decided on the reflection that on the next iteration we would include many different categories that we can include different products on the system.

**Iteration 2**

For the second iteration, we had already a clear idea in mind of what we needed to do to the application. We needed to add more categories and products to the system, and make it capable of handling more products. We concluded this from out sprint reflection in the previous iteration. However, we decided on a new goal for the end of this iteration, which was to allow the cashiers to test our system and to gain their insight into the system. We thought of using the think aloud observation technique.

**Development**

During development, we now needed to add a lot more categories to the system, and generally make the system easier, and now include the order system. Therefore, we first developed more categories, and we began increasing the optimization to handle more products on the system. We came across difficulty during the process due to the activities on android requiring to send a lot of data between each, meaning the system would crash and no longer work. To fix this we had to reduce the amount of data sent between activities, and reduce data sent to the database, as it could also not handle hash maps, which is what we first used to handle products of an order.

Once we had resolved this issue, we developed the order page and confirmation, which simple sent different orders to the database to be viewed by the shops when they access the database. Once we had implemented this feature, the new prototype was ready to be tested, so we set out and aimed to catch cashiers, hopefully when they were not busy, and ask them to use our app whilst talking aloud about what they thought and were doing.

**Think Aloud**

We decided that three from Conad and three from Carrefour was a nice, round number for us to be able to get some good data. Therefore, we first went to Carrefour.

***Carrefour***

*Cashier One*

The first cashier used the application, describing each action they were doing and describing what they think would happen. For the most part, they liked the application, and said they themselves would use it. However, he as a cashier was worried about any special offers within the store that would not be on the system. He suggested the idea of being able to add and remove products to the system.

*Cashier Two*

The second cashier seemed dismissive of the application, however when it came to conclude we found that he found it easy to use, he would accept it in the real world format, however he would like for edit/remove products also, and also said he would like to login as an administrator.

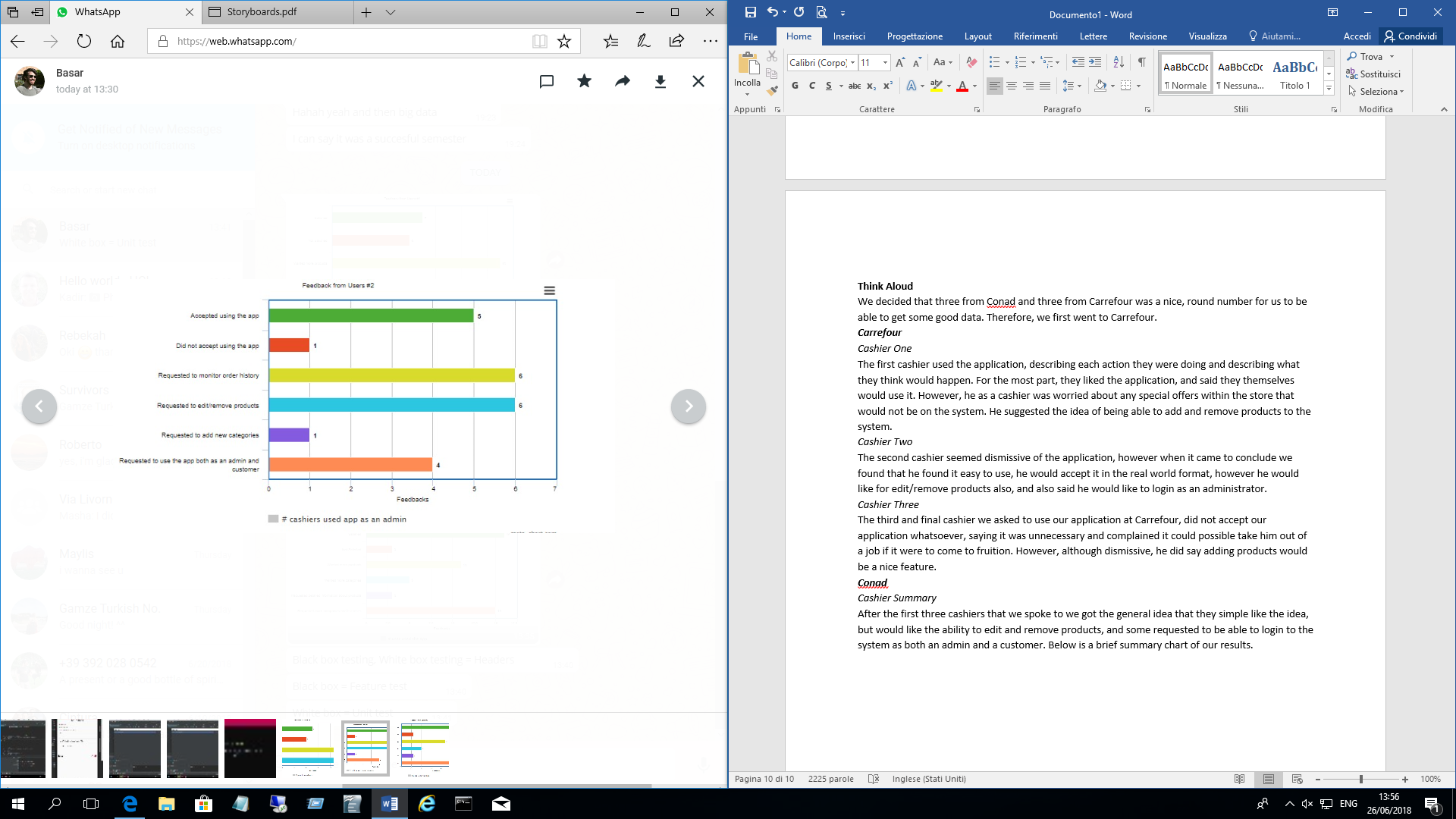
*Cashier Three*

The third and final cashier we asked to use our application at Carrefour, did not accept our application whatsoever, saying it was unnecessary and complained it could possible take him out of a job if it were to come to fruition. However, although dismissive, he did say adding products would be a nice feature.

***Conad***

*Cashier Summary*

After the first three cashiers that we spoke to we got the general idea that they simple like the idea, but would like the ability to edit and remove products, and some requested to be able to login to the system as both an admin and a customer. Below is a brief summary chart of our results.



**Summary of Iteration**

To summarize this iteration, we added more categories, and more products, and then took this application to the cashiers to gain their insight into our application. We found that many cashiers approve of this application, which is great because it is significantly better than our previous result, and it means we are heading into the correct location for our application. For a reflection of the sprint however, users requested the ability to add and remove products to their store, therefore, we will need to add this feature in the future.

**Iteration 3**

For this iteration, we decided that we need to implement the ability to add and remove products on the system, and would like to generally make the user interface flow easier. At the end of this iteration, we decided to go back to the user again, deciding to possibly stand outside a supermarket so we directly ask our real customers, and get their opinion. We aimed for roughly 20 interviews on the field.

**Development**

Development stage for this iteration was relatively straightforward, we simply needed to add an admin section in which users can edit and remove products from the system, and of course, we would need to make the use of this application easier for people to use to accommodate a wider audience.

**On-the-field Interviews**

We first allowed the user to carry out a think aloud exercise, as it proved useful with the cashiers. Therefore, the users carried out the exercise and spoke through each process, and a few ideas came from users in what they would like to change and see additionally. User still requested even more categories and products, and of “special offer” categories. Some also asked for detailed information of a product, such as nutritional value and its ingredients etc., and a lot asked for more companies and branches to be added to the system.

After this initial think aloud exercise we asked a few additional questions:

**Are you satisfied with the application?**

* 16 yes
* 3 No

**If yes, what do you like?**

***Summarized:***

* Simple
* Easy to use
* Many products to choose from

**If no, what did you dislike?**

***Summarized:***

* Want more categories
* Want detailed information for allergens

1. **Testing**

**Black Box – Feature Test / Basic Function Test**

**Scenario based**

In this part, we focused on some scenario to assess whether our software meets the requirements or not. Below there are nine scenario and their results.

**Scenario 1:** User registers into the system

**Goal:** For register, user enters name, surname, email, phone number, address, password, country, city, and region

**Result:** User enters all information and click submit button. Then, registers successfully.

**Scenario 2:** User logs into the system

**Goal:** For login, user enters email, password

**Result:** User enters email and password and clicks login button. Then, user logs in successfully.

**Scenario 3:** User orders product

**Goal:** After login, company page should be displayed. After selecting company, product list should be displayed. After selecting products order process should be completed successfully

**Result:** User logs in and selects company to see available products successfully. Then, after selecting products user clicks confirmation button to finish order process successfully.

**Scenario 4:** User monitors his/her order history

**Goal:** After login, user should open the navigation menu from upper left corner of the company page or products page. Then user should select the “order history” button to see previous orders.

**Result:** After successful login, user opens the navigation menu from upper left corner of the company page and selects the order history button. As a result, user can monitor order history successfully.

**Scenario 5:** User changes password

**Goal:** After login, user should open the navigation menu from upper left corner of the company page or products page. Then user should select the “Account Settings” button to see personal information and “change password button”. User should click “Account Settings” and fill old and new password. System should change the user password.

**Result:** After successful login, user opens the navigation menu from upper left corner of the company page. Then user selects the “Account Settings” button and can see personal information and “change password button”. User clicks “Account Settings” and fills old and new password. System changes the user password successfully.

**Scenario 6:** User changes address

**Goal:** After login, user should open the navigation menu from upper left corner of the company page or products page. Then user should select the “Account Settings” button to see personal information and “change address button”. User should click “Account Settings” and fill new address. System should change the user address.

**Result:** After successful login, user opens the navigation menu from upper left corner of the company page. Then user selects the “Account Settings” button and can see personal information and “change address button”. User clicks “Account Settings” and fills new address. System changes the user address successfully.

**Scenario 7:** Admin adds product

**Goal:** After login, user should open the navigation menu from upper left corner of the company page or products page. Then user should select the “Admin” button to see whether this user is an admin or not. After clicking “Admin” button, user should click “Add Product” and fill the information of product. System should add the product with given information to the selected company.

**Result:** After successful login, user opens the navigation menu from upper left corner of the company page or products page. Then user selects the “Admin” button and sees this user is an admin successfully. After clicking “Admin” button, user clicks “Add Product” and fills the information of product. System adds the product with given information to the selected company successfully.

**Scenario 8:** Admin edits product

**Goal:** After login, user should open the navigation menu from upper left corner of the company page or products page. Then user should select the “Admin” button to see whether this user is an admin or not. After clicking “Admin” button, user should click “Edit Product” and change the information of product. System should change the product with given information for the selected company.

**Result:** After successful login, user opens the navigation menu from upper left corner of the company page or products page. Then user selects the “Admin” button and sees this user is an admin successfully. After clicking “Admin” button, user clicks “Edit Product” and changes the information of product. System changes the product with given information to the selected company successfully.

**Scenario 9:** Admin removes product

**Goal:** After login, user should open the navigation menu from upper left corner of the company page or products page. Then user should select the “Admin” button to see whether this user is an admin or not. After clicking “Admin” button, user should click “Remove Product” and remove the selected product. System should remove the product from the selected company.

**Result:** After successful login, user opens the navigation menu from upper left corner of the company page or products page. Then user selects the “Admin” button and sees this user is an admin successfully. After clicking “Admin” button, user clicks “Remove Product” and removes product. System removes the product from the selected company successfully.

**White Box Testing**

**Unit test**

**Function 1:**

**TestisEmailValid() LoginActivity.kt**

Tests isEmailValid(email: String) : Boolean method

Test 1:

Input: testmail.com Returned: True Expected: False **Action: Function fixed**

Test 2:

Input: testmail.com Returned: False Expected: False

Test 3:

Input: [test@mail.com](mailto:test@mail.com) Returned: True Expected: True

**Verdict: Function validated**

**Function 2:**

**TestisPasswordValid() LoginActivity.kt**

Tests isPasswordValid (password: String) : Boolean method

Test 1:

Input: pas Returned: False Expected: False

Test 2:

Input: pass12345 Returned: True Expected: True

**Verdict: Function validated**

**Function 3:**

**Test**checkAddress**() ChangeAddressActivity.kt**

Tests checkAddress (email: String) : Boolean method

Test 1:

Input: null Returned: False Expected: False

Test 2:

Input: “Test Address 2” Returned: True Expected: True

**Verdict: Function validated**

**Function 4:**

**Test**checkOldPassword **() ChangePasswordActivity.kt**

Tests checkOldPassword(pass : String) : Boolean method

Test 1:

Input: Different pass. Returned: False Expected: False

Test 2:

Input: Same pass. Returned: True Expected: True

**Verdict: Function validated**

**Function 5:**

**Test**calculatePrice **() GridAdapter.java**

Tests Double calculatePrice(String totalPrice, Double productPrice, Boolean Increment) method

Test 1:

Input: (10, 5, True) Returned: 15 Expected: 15

Test 2:

Input: (10, 5, False) Returned: 5 Expected: 5

**Verdict: Function validated**

**Function 6:**

**Test**checkEmail **() RegisterActivity.kt**

Tests isPasswordValid(password: String) : Boolean method

Test 1:

Input: testmail.com Returned: False Expected: False

Test 2:

Input: [test@mail.com](mailto:test@mail.com) Returned: True Expected: True

**Verdict: Function validated**

**Function 7:**

**Test**checkNum**() RegisterActivity.kt**

Tests checkNum(num : String): Boolean method

Test 1:

Input: null Returned: False Expected: False

Test 2:

Input: “324 548 8089” Returned: True Expected: True

**Verdict: Function validated**

**Function 8:**

**TestcheckNames() RegisterActivity.kt**

Tests checkNames(firstname : String,lastname : String): Boolean method

Test 1:

Input: null, “Kamil” Returned: False Expected: False

Test 2:

Input: “Kamil”, null Returned: False Expected: False

Test 3:

Input: “Kamil”, “Kamil” Returned: True Expected: True

**Verdict: Function validated**

**Function 9:**

**TestcheckPassword() RegisterActivity.kt**

Tests checkPassword(password1 : String,password2 : String): Boolean method

Test 1:

Input: “pass123”, “pass223” Returned: False Expected: False

Test 2:

Input: null, null Returned: Null Error Expected: False **Action: Function fixed**

Test 3:

Input: null, null Returned: False Expected: False

Test 4:

Input: “pass”, “pass” Returned: False Expected: False

Test 5:

Input: “pass1234”, “pass1234”

Returned: True Expected: True

**Verdict: Function validated**

**Function 10:**

**Test**getSingleProduct**() Utilities.kt**

Tests getSingleProduct(key: String, listener: FireBaseListener) method

Test 1:

Input: product key, listener Result: Correct product returned

Test 2:

Input: null, listener Result: No product returned

**Verdict: Function validated**

**Function 11:**

**Test**getSingleProduct**() Utilities.kt**

Tests getSingleProduct(key: String, listener: FireBaseListener) method

Test 1:

Input: (product key for Coca-cola, listener) Result: Correct product returned on callback

Test 2:

Input: null, listener Result: No product returned

**Verdict: Function validated**

**Function 12:**

**Test**getSingleCompany**() Utilities.kt**

Tests getSingleCompany (key: String, listener: FireBaseListener) method

Test 1:

Input: (company key for Carrefour, listener) Result: Correct company returned on callback

Test 2:

Input: null, listener Result: No company returned

**Verdict: Function validated**

**Function 13:**

**Test**updateProduct**() Utilities.kt**

Tests updateProduct(product: Product) method

Test 1:

Input: Product class for Coca-cola with updated price Result: Price updated

Test 2:

Input: Product class for Coca-cola with updated name Result: Name updated

Test 3:

Input: Product class for Coca-cola with updated image Result: Image updated

Test 4:

Input: Product class for Coca-cola with new company Result: Company updated

Test 5:

Input: null Result: Nothing updated

**Verdict: Function validated**

**Function 14:**

**Test**updateCompany**() Utilities.kt**

Tests updateCompany(company: Company) method

Test 1:

Input: Company class for Carreffour with updated hours Result: Hours updated

Test 2:

Input: Company class for Carreffour with updated name Result: Name updated

Test 3:

Input: null Result: Nothing updated

**Verdict: Function validated**

**Function 15:**

**Test**updateOrder**() Utilities.kt**

Tests updateOrder(orderKey: String) method

Test 1:

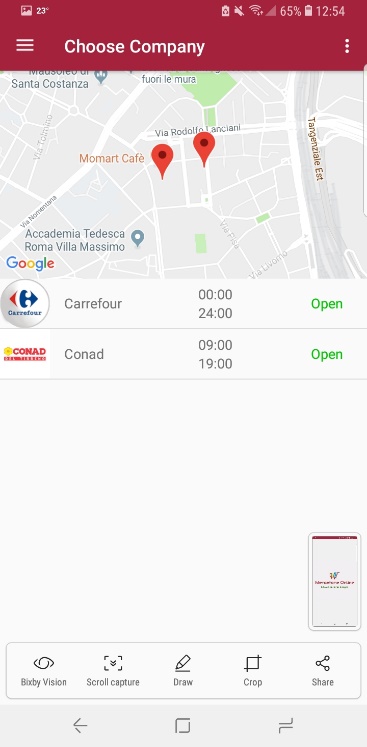
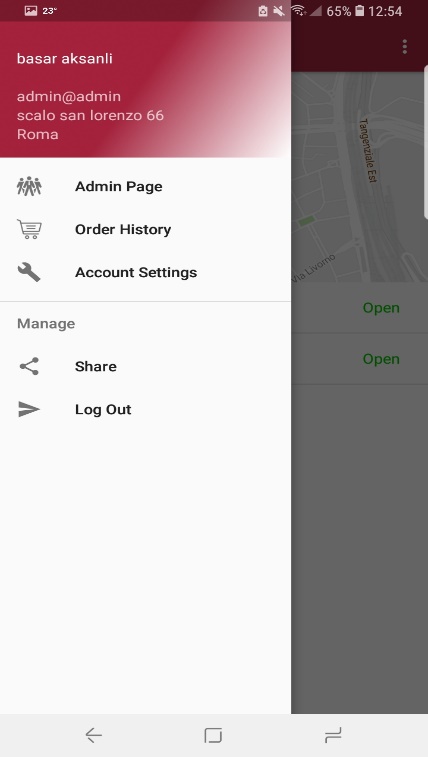
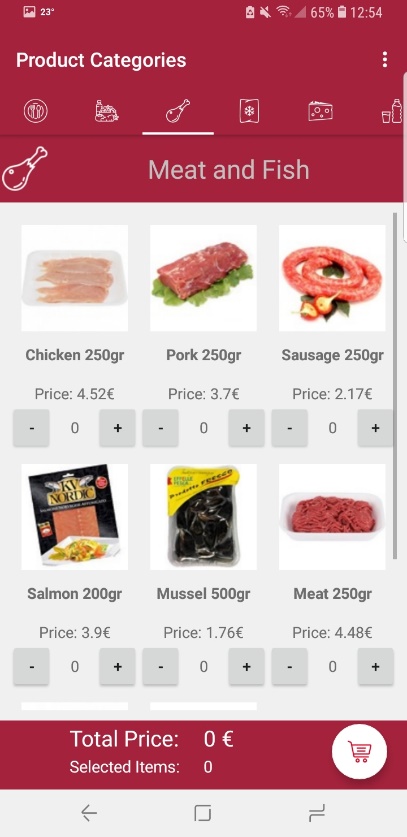
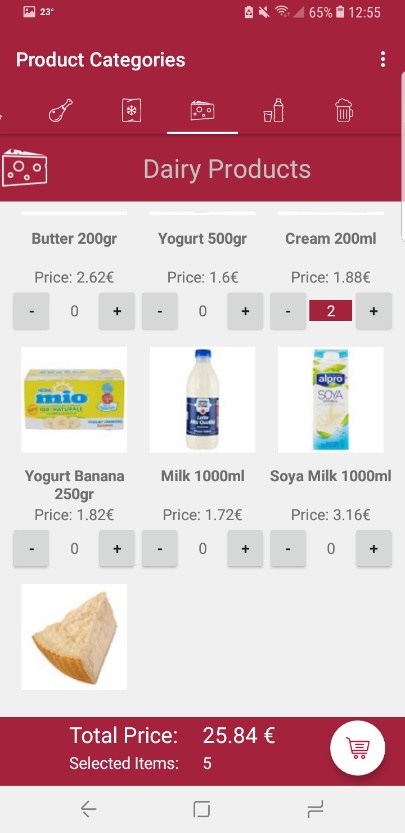
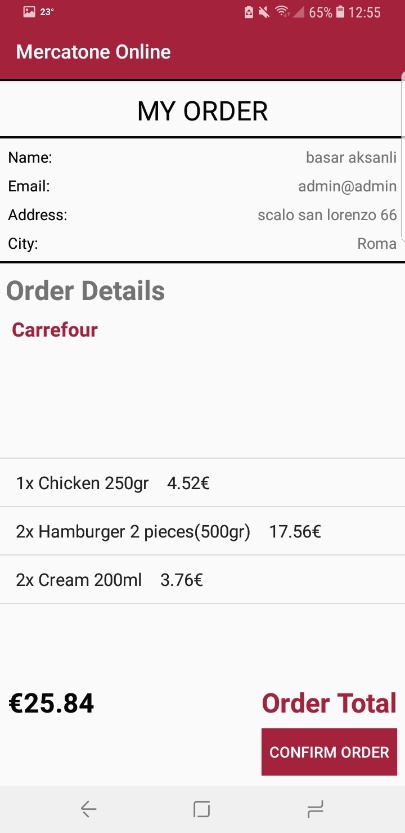
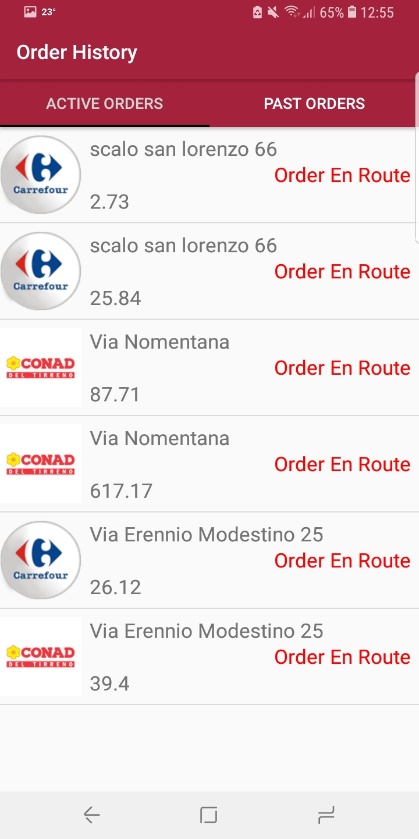
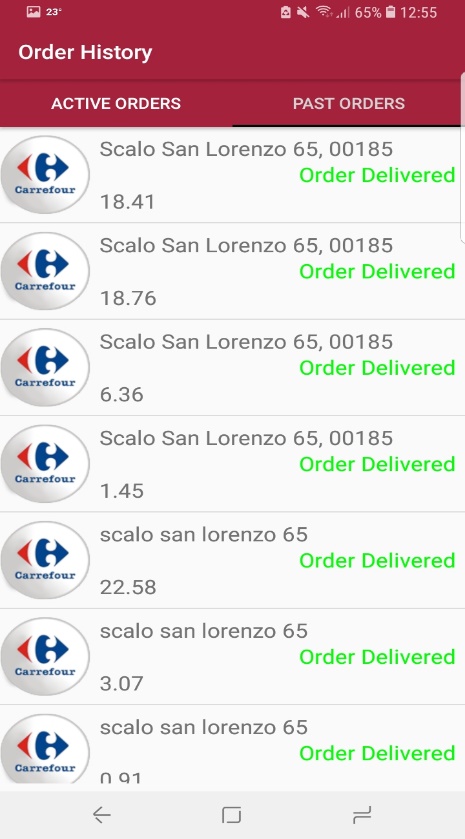
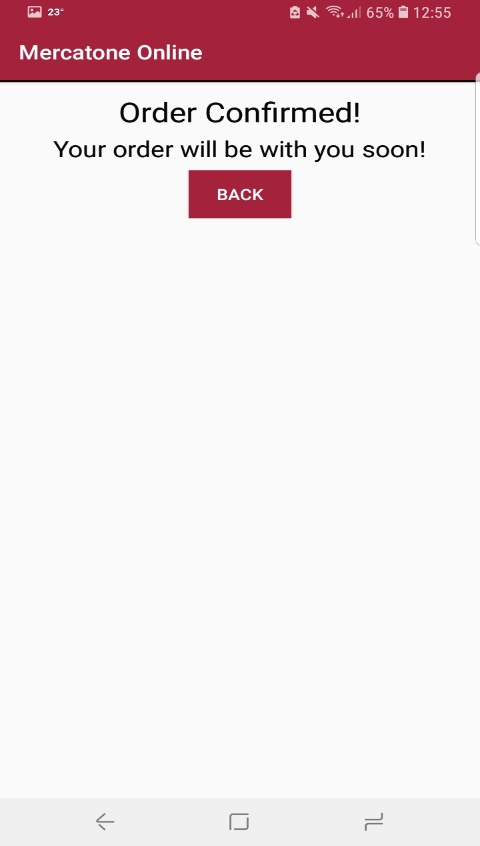
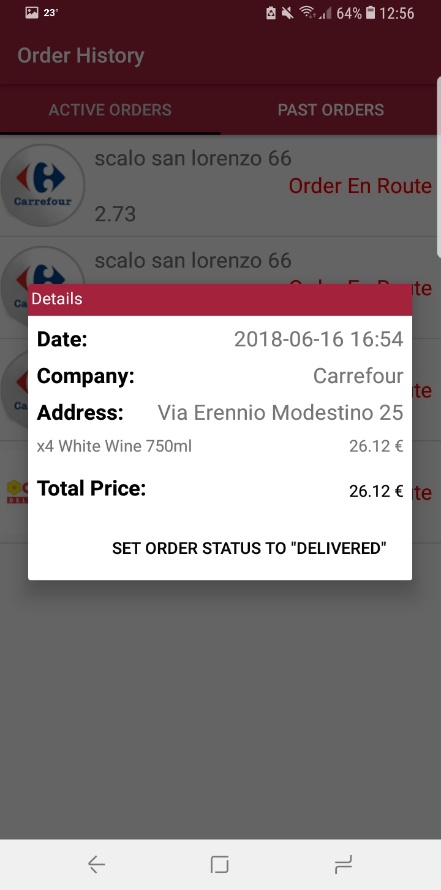
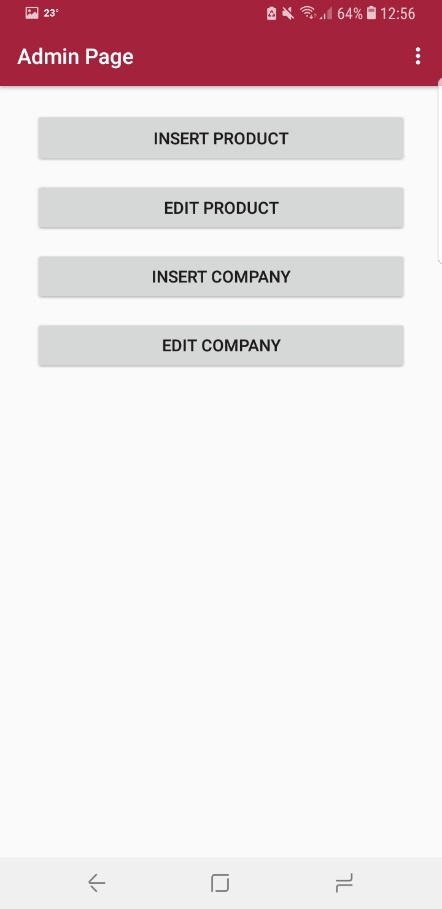
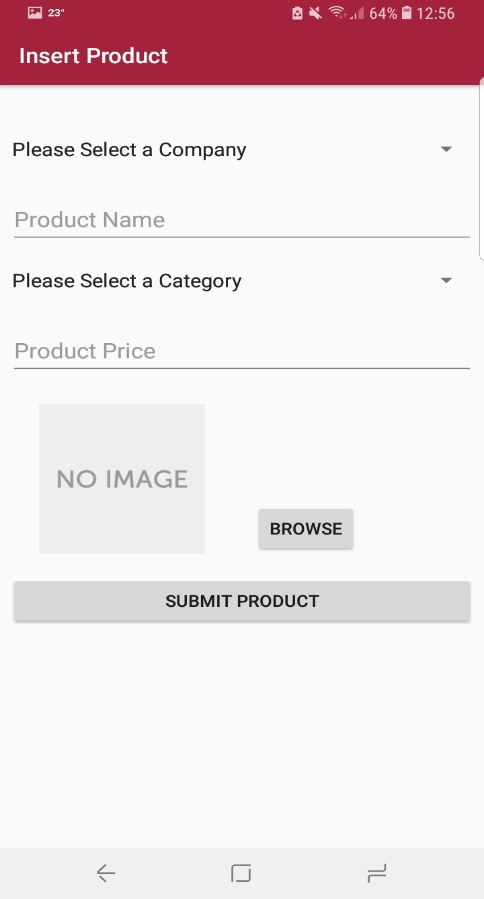
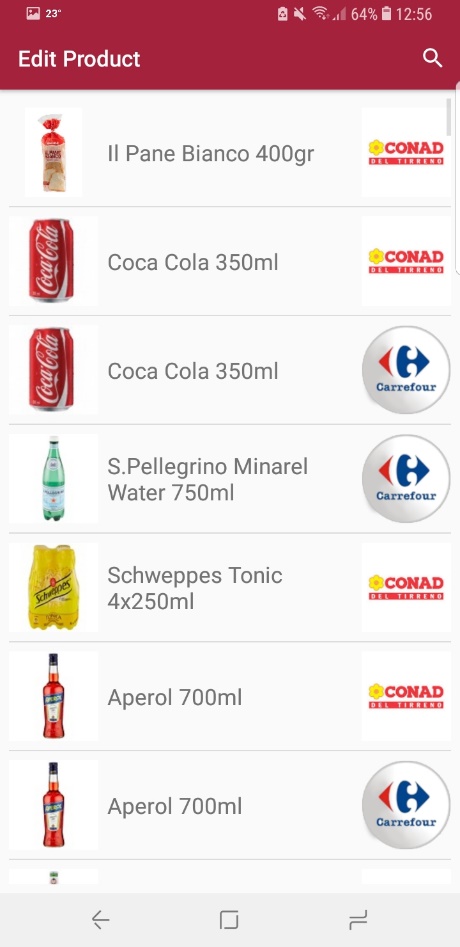
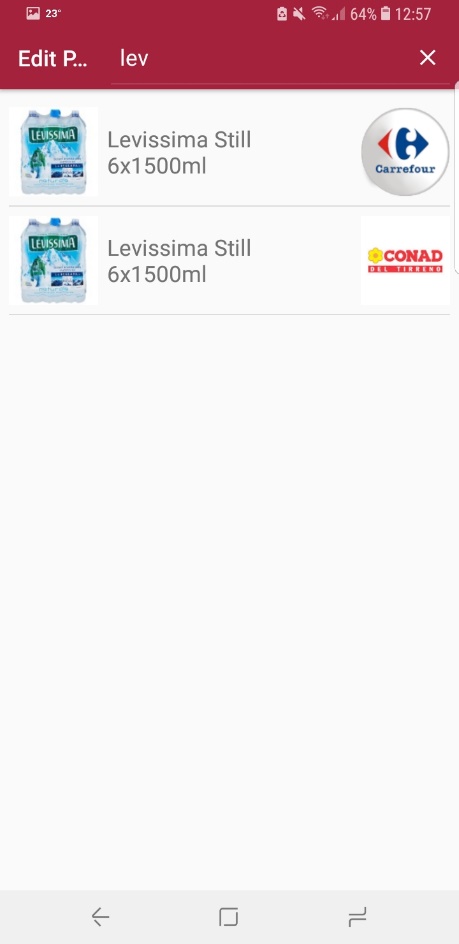
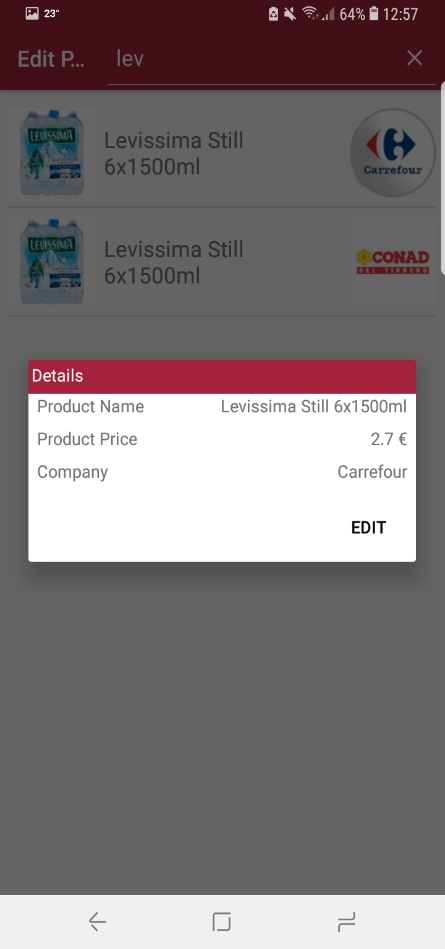
Input: OrderKey Result: Order status set to “Done”

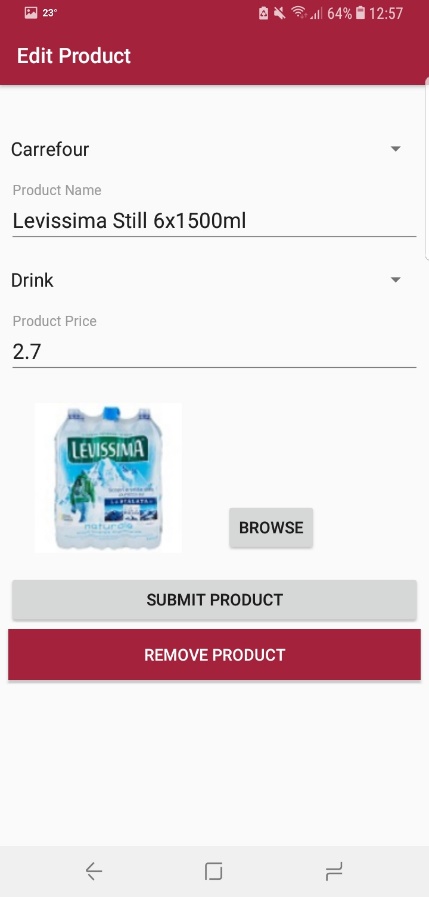
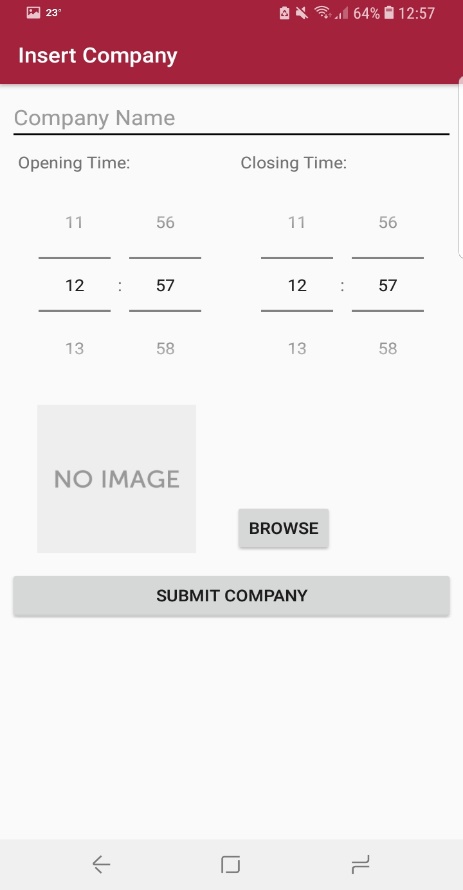
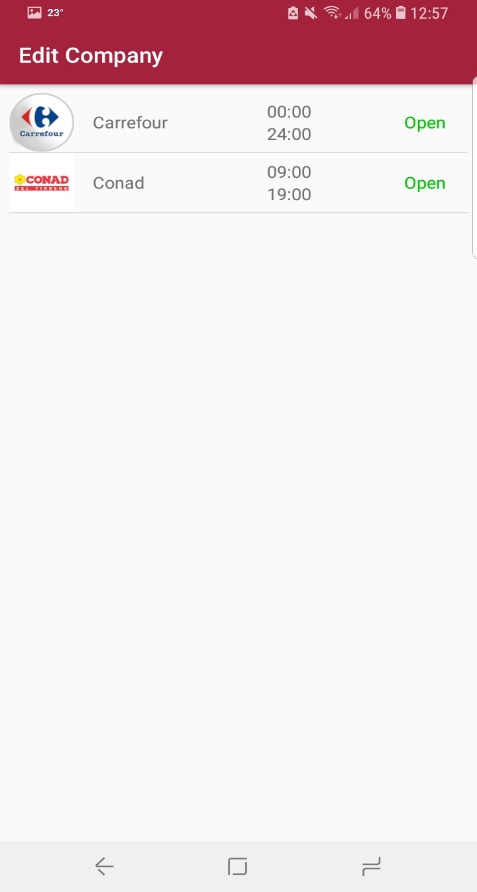
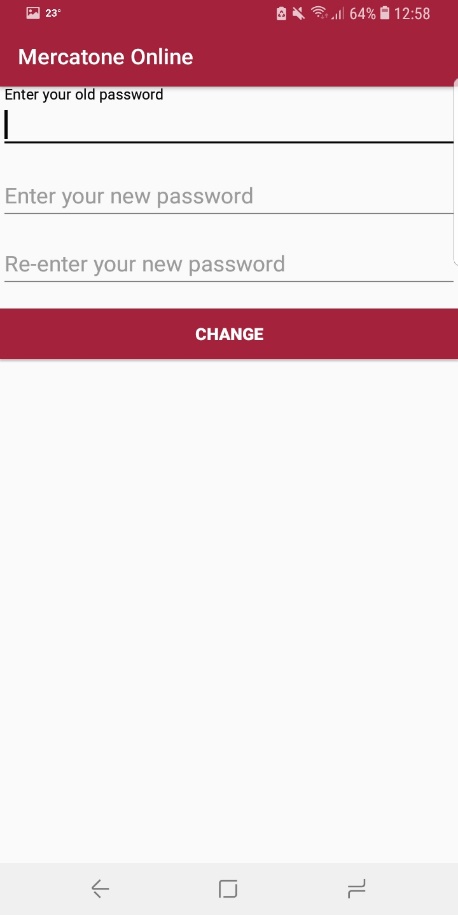
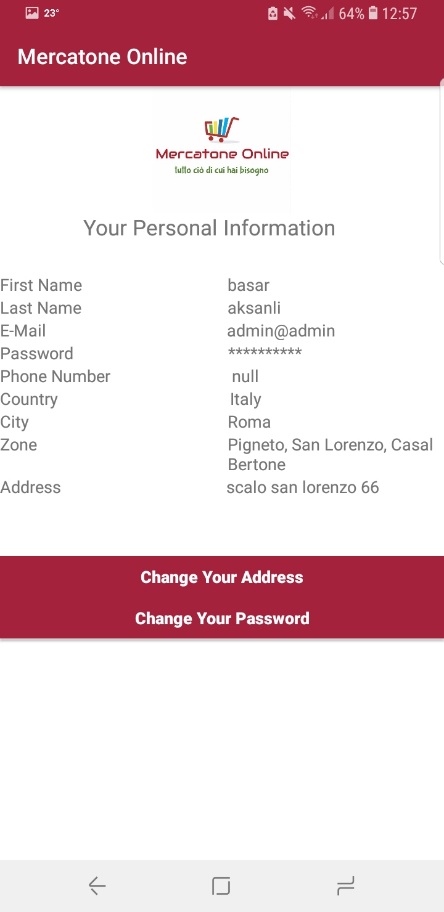
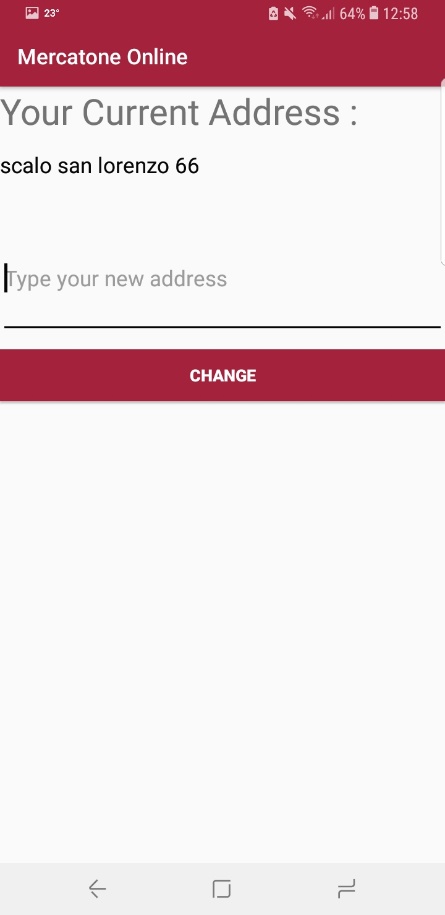
Test 3:

Input: null Result: Nothing updated

**Verdict: Function validated**

1. **Final Project**

**** ****       **** ** **    

1. **Conclusion**

To conclude, we completed an application in which a user can shop from a supermarket from within this application. We found that to begin with, the user perception of such an idea was not so great but we improved upon this on each iteration, until we got to our final project, which is shown above. We have found that people like the idea of the final project and are very welcoming of the idea we presented, we believe if this were to be further developed in the future, it would become a success on the app store.