UNIVERSITÀ DEGLI STUDI DI MILANO
FACULTY OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

Master in
Computer Science

SIMULATION
FINAL REPORT ON AGENT-BASED SIMULATION FOR STIROAPP

Teacher: Prof. Alberto Ceselli

Final report written by:
Samuele Simone
Matr. Nr. 11910A

Academic year 2022-2023

Contents

Chapter 1

Introduction

The idea behind this project was to try to validate a startup idea by making use of simulation models. Specifically by leveraging an agent-based simulation. In the report therefore we will go over what StiroApp is, how it works, the models that have been used as well as the technical implementation. Through the use of KPIs we will study the performance indicators of the model. After that there will be a chapter dedicated to experiments and finally the conclusions that can be drawn from it.

1.1 StiroApp - The story behind

StiroApp was born from the desire to bring to the market of applications, and more generally of services, a system that can connect two categories of people: Workers and Buyers. In more detail, the service seeks to make the process of ironing/washing one's clothing faster, more streamlined and more economical. I started by analyzing people's desire/problems by going to create user-stories necessary to make the product as user-centric as possible. As an example I reported one user story for a better understanding of the problem.

1.1.1 User story - Max

- **Context**: Max is a 27-year-old boy. He has been living alone for about a year. He moved to Milan for work reasons. He's a video game enthusiast.
- **Problems:**
  - Because of his busy life, He does not have time to iron/wash his clothes
  - It is a time-consuming, tedious and even difficult activity

1.2 Understanding the problem - More deeply

After a series of dutiful analyses and interviews at the front to examine the problem as best as possible here is what can be extracted in a more abstract way representing the basic principles for which is Stirapp should be born.

- Only a few millennials make use of laundries
- Daily laundry/ironing cannot be delegated to laundries
- This type of daily activity (ironing/washing) is generally carried out by generation X (1965 - 1980) as opposed to generation Y (1980 - 1994) and Z (1995 - 2010)
- Delivery/collection times of clothes are high

1.3 Proposed solution

The approach taken to solve these problems was therefore to offer an application for people in such a way as to connect the two categories into which we can classify users:

- Workers: Those who intend to use the application for the purpose of earning money by working for Buyers, thus ironing or washing their clothes.
- Buyers: Those who intend to use the app as a service where they can get their laundry cleaned and ironed.

1.3.1 Starting features

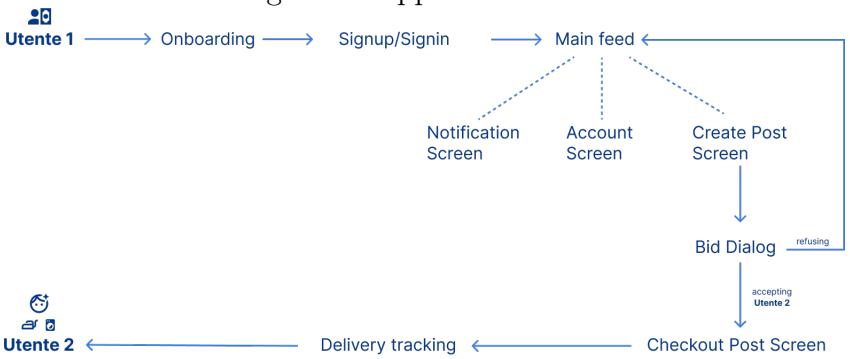
The app can be divided into two major features:

- Create posts for your own clothes by specifying the type of service you want (example: washing, ironing, both) and your needs.
- Wash, iron other users' clothes in such a way as to earn money.

1.3.2 App Flow

Below I show how the app should work. As an example during the agent-based simulation, some changes were made so as to focus more on the important parts of the app such as post creation or the worker system.

Figure 1: App flow Schema



- **Utente 1 (Buyers)** enter in the app after the onboarding in which he can understand the main features of the app
- **In-app onboarding** is the process of teaching users how to use an app to achieve their goals
- **Signup-Signin**. The user must be logged or registered into the system before using the app
- The user will be redirect into the Main Screen of the app. Then with a bottom navigation bar he can choose in which page enter such us Notification screen, Account screen and the Create Post screen.
- **Bid system** is a hypothetical part of the app in development
- **Checkout post screen** is the confirm page after the accepted price
- **Delivery tracking** is an important page for the user in order to check where is his order
- **Utente 2** he is the Worker that is allowed to do the laundry or offer other type of services specified in the app during the create post screen

1.3.3 UX and UI for StiroApp

Before we get our hands on the code, it is important to study the user experience (UX) and user interface (UI) to finalize the creation of a user-friendly application. So here I report some images of the work done:

Figure 2: Representation of the main features of the application through UX

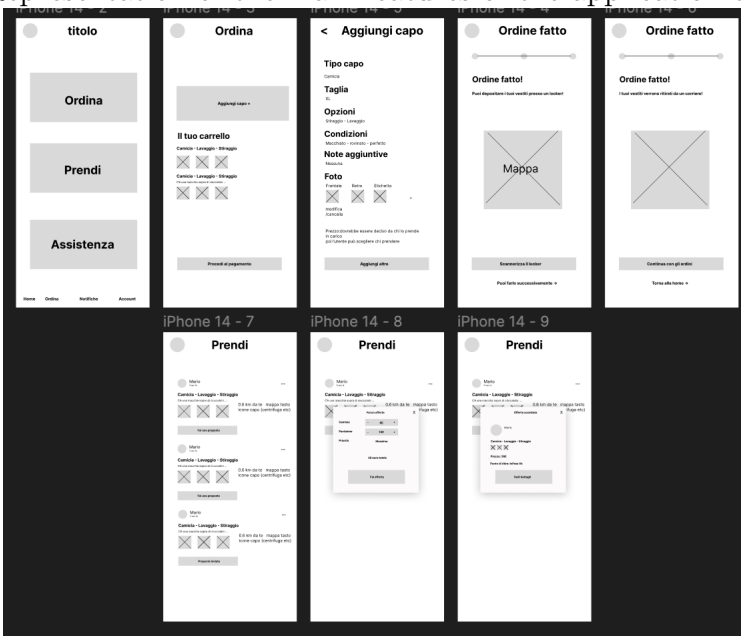


Figure 3: An example of Card UI for the post creation

1.3.4 Mockup

To give a real representation of StirApp to the reader I created a mockup. According to Decode Agency [1], an app mockup is a detailed representation of your app design. It contains all the final UI elements such as typography, copy, colors, and visuals like icons and photos. Sample content will also be used, but dummy text is also acceptable.

Figure 4: Main Feed Mockup

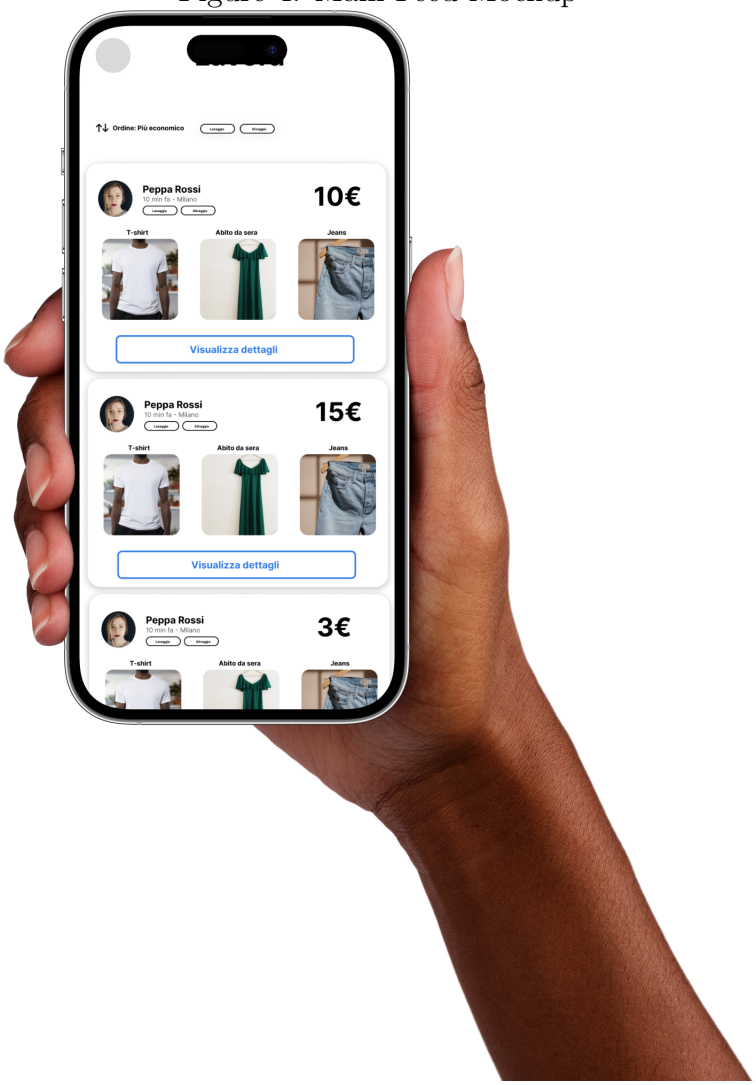


Figure 5: Detail Item Screen Mockup

1.4 A look into the market

By doing a market analysis I obtained those main pillars:

- Buyers find this activity (washing/draining) time-consuming
- Workers find this activity (washing/ironing) necessary and have the skills to do it
- Buyers prefer to find someone who can take care their clothes
- Workers make a profit for fulfilling these tasks

1.5 Business Model

To conclude this introductory part and to make sure that the project can have an economic source of revenue we need to proceed in writing a business model. My business model is based on the number of valid transactions within the application. A valid transaction is when the work is completed by both parties.

- Fixed cost + variable percentage based on post cost paid by buyers.

- Applicable for every valid transaction
- In the future, cleaning products (internal e-shop)
- Advertising in some list item (non-invasive)

Currently within the simulation only point 1 and 2 has been carried out. The remaining ones, on the other hand, are to be applied in such a way as to try to increase revenue more and have a higher margin than the costs incurred. Costs are defined as the following:

- Infrastructure costs of the service
- Costs of team management and administration.
- Costs on privacy and damage of garments

Chapter 2

Model

In order to simulate the traffic over the app and to estimate revenues, costs and other KPI that are described in the Chapter 4, I used the agent-based modeling. As well described from the Columbia University website [2], Agent-based models are computer simulations used to study the interactions between people, things, places, and time. They are stochastic models built from the bottom up meaning individual agents (often people in epidemiology) are assigned certain attributes.The agents are programmed to behave and interact with other agents and the environment in certain ways.

The agents that are involved in the model are:

- **Worker_iron**: Represents the worker within the application
- **Buyer**: Represents the buyer, i.e., the one who creates the posts with their clothing
- **Rider**: He is in charge of picking up the goods and delivering them to the respective worker/buyer.
- **Scheduler**: He/she is in charge of handling incoming posts and directing them

We will see during this Chapter how I modeled these agents for StiroApp.

2.1 Worker_iron agent

Below I will go on to describe the logic of the Worker_iron agent. The technical details will be discussed in Chapter 3. Basically the worker as soon as it is generated waits for some order to be assigned to it, in a **Waiting state**. Through a branch the worker by condition wonders whether it has received the order or not.

In the first case then he can start working by entering the **Working state** and after a Uniform Discrete random variable $\mathcal{U}(a, b)$ where $a = 2, b = 60$ minutes, he enter in a **ReadyForDelivery state** Then, with a 10 minutes Timeout transition enter in a branch and through a Bernoulli random variable $\mathcal{B}(p)$ where $p = 0.7$ he decides whether to finish his task or re-enter the **Waiting state**.In the second case, on the other hand, he enters the **Thinking state** and then after a small timeout point to the same transition branch where there is the Bernoulli random variable.

2.2 Buyer agent

The Buyer as soon as it is generated is in a **Waiting state**. In fact here, through a transition to the branch you want to check if that specific agent is new or already created and therefore is waiting for his order. If he is new then he can create a post with his clothing by entering the **CreatePost state**. After that through a Timeout of 10 min he enters the **Waiting state** and also through a Bernoulli random variable $\mathcal{B}(p)$ where $p = 0.7$ he decides whether to finish his task or re-enter the **Waiting state**. If,instead, he has already a pending order and it's notified as delivered he can again through the same branch if finish or re-enter.

2.3 Rider agent

The rider plays a key role within my simulation. In fact it involves a slightly more complex logic in that its task depends on the type of order received. Initially it is in a **Waiting state**. As soon as it receives an order it goes into the **AssignedOrder state**. Then it has to figure out through the properties of the received order whether it is a pickup or a delivery. In fact the rider first has to go to the buyer and receive the clothes for which it wants to perform a service and from there, the rider has to deliver it at the chosen worker. Once this is done through a Bernoulli random variable $\mathcal{B}(p)$ where $p = 0.7$ decides whether to continue working enter the Waiting state again or stop working. If it continues working it may happen to receive the delivery order and then go to a worker who is in the ReadyForDelivery state, pick up the clothes, and return them to the buyer who owns those clothes.

2.4 Scheduler agent

This agent, on the other hand, acts as a conduit between agents by handling the exchange of messages (specifically, we will see that we will be dealing with the dispatch of an object of type Post). It presents two states, a **Wait state** and

a **Dispatch state**. It simply presents two transitions and allow you to manage the event queue and based on the type of post received sort the dispatches to the correct recipient.

Chapter 3

Implementation

ci

Chapter 4

KPI

ci

Chapter 5

Experiments

Chapter 6

Conclusion

ci

Bibliography

[1] Decode Agency. What is an app mockup? https://decode.agency/article/app-mockup/#:~:text=with%20proper%20wireframing-,What%20is%20an%20app%20mockup%3F,dummy%20text%20is%20also%20acceptable., 2022.

[2] Columbia University. Agent-based modeling. https://www.publichealth.columbia.edu/research/population-health-methods/agent-based-modeling#:~:text=Agent%2Dbased%20models%20are%20computer,epidemiology)%20are%20assigned%20certain%20attributes., 2023.