

# Final Project Detailed Instructions

---

## 1 Software Requirement Specification

### 1. Introduction

1. Aim of the document, I
2. Overview of the defined system
3. HW e SW requirements
4. Related systems (at least 2), Pros and Cons.

### 2. User Stories

1. Da ristorante, voglio poter mostrare le immagini dei prodotti, in modo da rendere più accattivante il menù.
2. Da ristorante, voglio illustrare gli ingredienti dei miei piatti, così da facilitare i clienti ad ordinare i piatti che incontrano i loro gusti.
3. Da cliente, voglio che mi venga mostrato il conto totale prima di pagare, in modo da farsi un'idea sull'importo totale.
4. Da cliente, voglio che mi venga mostrato il conto suddiviso per il numero di persone al tavolo, così da velocizzare il pagamento.
5. Da ristorante, voglio mostrare il mio menù diviso in categorie, in modo da rendere più facile ordinare per i clienti.
6. Da ristorante, voglio poter modificare il menù, così da essere più flessibile sul mercato.

### 1. Functional Requirements

1. Il sistema deve mostrare immagini selezionabili dei prodotti, accompagnanti dalle informazioni inerenti al singolo prodotto\*.
2. Il sistema deve offrire, all'utente "ristoratore", la possibilità di aggiungere e togliere elementi della lista dei prodotti mostrati.
3. Il sistema deve offrire, all'utente "ristoratore", la possibilità di modificare le informazioni inerenti al singolo prodotto\*.

4. Il sistema deve sommare i prezzi dei prodotti ordinati.
5. Il sistema deve offrire una casella di testo, per ogni piatto presente sul menù, utilizzata per elencare gli ingredienti che compongono il piatto.
6. Il sistema deve visualizzare, all'utente "cuoco", tutti gli ordini effettuati.

7. Use Cases:

1. Overview Diagram (1)
2. Internal Steps ( 1 per member)

## 2 Storyboards

Quantity: min 2 screens per member, covering all the major functionalities described in SRS, developed using Draw.io or similar

## 3 Design

- Class diagram:
  - 1 VOPC per member.(analysis) ◦ 1 design-level diagram per member (e.g. that includes patterns, or specific solutions that improve the engineering level of the system)
- Design patterns: 1 different pattern per member. Possibly try to apply the pattern within the context of the project.
- Activity diagram: 1 per member.
- Sequence diagram: 1 per member.
- State diagram: 1 per member.

## 4 Testing

- Develop at least 3 test cases per person. In each test (class) file, please report (via Java comments) the name of the person in charge.
- 1 Selenium test via GUI per member.
- 1 Selenium test via API per member.

## 5 Code

- ~4K LOC
- Similar functionality implemented with GUIs.

- Exceptions: at least 2 per member (do not just catch and back-propagate the exceptions, but properly handle them. Possibly define your own error logic by means of exceptions)
- Be able to show that Svn(or Git) + SonarCloud is correctly installed in one of your computer and it is able to analyze your project for rule violations. No rule must be violated (no smells, no vulnerabilities, no bugs). This will be checked during the exam.
- One DAO shall be provided in two versions DMBS and file system.

## 6 Video

- A 1 to 2 minutes recorded video with audio of the developed system performing the expected functionalities. \*.mpeg

### Notes

1. System.out\* CLI-related smells are allowed.
2. Deliverables should be compliant with HMW constraints.
3. Code must be compliant with deliverables. Discrepancies can exist and they should be pointed out in such artifacts.
4. The project is one per team. The grade is shared across team members. All team members must submit the entire project.
5. The project submission procedure will be communicated (only) to the students subscribed to a specific exam date.