

ReserveWell	V1.3
Iteration Plan	Date: 2/12/2023

# ReserveWell

## Second Iteration Plan-Revised

### Version History Table

Version	Date	Description
v1.0	4.11.2023	-
v1.1	8.11.2023	Data base and "attrbiutes" are corrected to "database" and "attributes."
v1.2	25.11.2023	<ul style="list-style-type: none"> <li>• Work Items List is updated.</li> <li>• The statuses in the Issues section are updated.</li> <li>• New evaluation criteria have been added.</li> <li>• The subheadings in the Assessment section were rewritten according to the final version of the iteration.</li> <li>• Other concerns and deviations section is updated.</li> </ul>
v1.3	2.12.2023	<ul style="list-style-type: none"> <li>• Version number is added to the document.</li> <li>• Work Item List is updated.</li> </ul>

## 1. Key milestones

Milestone	Date
Iteration starts	6/11/2023
Define other use cases	8/11/2023
Define objects, attributes, and methods	14/11/2023
Determine initial database relationships	20/11/2023
Define initial software architecture	23/11/2023
Design initial user interface requirements	23/11/2023
Revise work list	23/11/2023
Evaluate other teams	26/11/2023
Get feedback from other teams	28/11/2023
Iteration stops	28/11/2023

## 2. High-level objectives

- **Define Objects, Attributes, And Methods:**

ReserveWell	V1.3
Iteration Plan	Date: 2/12/2023

Identify and document the objects (entities) within the system, including their attributes and methods. Objects represent the key components or entities in the system and are essential for modeling system behavior and data interactions.

- **Determine Initial Database Relationships:**

Establish the foundational database relationships required for the project, including the definition of tables, their attributes, and the relationships between them. Ensure that data can be stored and retrieved efficiently.

- **Define Initial Software Architecture:**

Create an initial software architecture that outlines the high-level structure and components of the system. This architecture should define how different modules or components will interact, ensuring scalability, maintainability, and performance.

- **Design Initial User Interface Requirements:**

Develop the initial user interface (UI) requirements, including wireframes, layouts, and user interaction flows. Ensure that the UI design aligns with the project's objectives and provides an intuitive and engaging user experience.

### 3. Work Item assignments

Name / Description	Priority	Size estimate (Points)	State	Reference material	Target iteration	Assigned to (name)	Hours worked	Estimate of hours remaining
Define roles and responsibilities	2	2	Assigned		2	Ezgi E.	2	6
Define system-wide requirements	2	3	Assigned	<SystemWideRequirements.pdf>	2	Mustafa	8	8
Develop a use-case models	2	5	Closed	<Use Case Diagram_ReserveWell>	2		6	15
Form use-case diagrams	1	3	Closed		2	Tunali	0	6
Define diner uses cases	2	13	Closed		2		11	29
Register	2		Closed	<UC1_Register(Diner).pdf>	2			
Exit from a Waitlist	2		Closed	<UC4_Exit From the Waitlist.pdf>	2	Ezgi E.	5	5
Make Reservation upon Waitlist Availability	2		Closed	<UC5_Make Reservation upon Wishlist Availability.pdf>	2	Ezgi E.		
Rate the Restaurant	2		Closed	<UC6_Rate the Restaurant.pdf>	2	Ezgi E.	3	3

ReserveWell	V1.3
Iteration Plan	Date: 2/12/2023

<b>Define restaurant manager uses cases</b>	2	13	Closed		2		0	29
RegisterR	2		Closed	<UC7_Register(Restaurant Manager).pdf>	2	Ezgi E.	0	6
Display Reservations	2		Closed	<UC8_Display Reservations (Restaurant Manager).pdf>	2	Tunalı	0	6
Manage Reservations	2		Closed	<UC9_Manage Reservations.pdf>	2	Tunalı	0	8
Edit Working Hours	2		Closed	<UC10_Edit Working Hours.pdf>	2	Tunalı	0	6
Edit Capacity	2		Closed	<UC11_Edit Capacity.pdf>	2	Tunalı	0	5
Display Dashboards	2		Closed	<UC12_Display Dashboards.pdf>	2	Tunalı	0	5
Manage Restaurant's Waitlist	2		Closed	<UC13_Manage Restaurant's Waitlist.pdf>	2	Ezgi E.	0	5
<b>Define waitstaff use cases</b>	2	13	Closed		2	Ezgi E.	0	27
Register	2		Closed	<UC14_Register (Waitstaff).pdf>	2	Ezgi E.	0	5
Display Reservations	2		Closed	<UC15_Display Reservations (Waitstaff).pdf>	2	Tunalı	0	6
Display Dashboards	2		Closed	<UC12_Display Dashboards.pdf>	2	Tunalı	0	8
Confirm Realized Reservations	2		Closed	<UC16_Confirm Realized Reservations.pdf>	2	Tunalı	0	8
<b>Support basic creating new reservation process</b>	2	8	Verified		2	Umut	3	24
Prototype UI	2		Closed		2	Umut	0	5
Do the design	2		Closed	<Design.pdf>	2	Umut	0	8
Implement	2		Closed	<Implementation.pdf>	2	Umut	0	5
Test			Assigned	<TestCases.pdf>, <TestScripts.pdf>	2	Umut	3	3
Verify data flow	2		Closed		2	Umut	0	3
Form dataflow diagram	2		Closed		2	Tunalı	0	3
Decide whether outsourcing or inhouse	3		New		2		8	8
Define design artifact	2		Closed		2			
Define implementation artifact	2		Closed		2			

ReserveWell	V1.3
Iteration Plan	Date: 2/12/2023

Define Iteration Plan 2	2		Closed	<IterationPlan2.pdf>	2	Ezgi E.	0	6
<b>Define offline scenarios for internet failures</b>	2	3	Closed (Out of Scope)		2		8	8

ReserveWell	V1.3
Iteration Plan	Date: 2/12/2023

## 4. Issues

Issue	Status	Notes
Order of the restaurants shown on the interface	Open	It cannot conclude yet whether restaurants should be displayed on the screen in alphabetical order, according to location information, or according to restaurant type.
How to share tasks among development team	Closed	
Any additional features needed for the core reservation system	Open	In the detailed design phase, the need for additional features may rise.

## 5. Evaluation criteria

- At the end of second iteration minimum%70 of the use cases should be written as fully dressed.
- Completeness and relevance of identified use cases, and alignment of use cases with project scope and objectives.
- No missing relationship in the database.
- Clarity and robustness of the initial software architecture.
- Walkthrough of iteration build with Quality Teams received favorable response. Also, minimum 85% of the opinions against the interface design should be positive.
- Quality and intuitiveness of the initial user interface design, and alignment with user experience objectives.
- Completeness and accuracy of project documentation.
- Clarity and completeness of object definitions, including attributes and methods, and alignment of object definitions with project requirements.
- To ensure that the reservations made by the memoirists through the application are instantly updated in the database.
- Successful completion of specified test scenarios. The detailed information related to test scenarios can be seen in the Test Cases, Test Scripts, Test Logs documents.

ReserveWell	V1.3
Iteration Plan	Date: 2/12/2023

## 6. Assessment

Assessment target	Second Iteration
Assessment date	28/11/2023
Participants	Development Team, Quality Team
Project status	Green

- **Assessment against objectives**

***Define Objects, Attributes, And Methods:***

The identification of objects appears comprehensive, covering key components of the system.

Attributes and methods are well-defined, providing clarity on the properties and behaviors associated with each object.

Documentation quality is satisfactory, offering clear information for developers.

***Determine Initial Database Relationships:***

Relationship definitions accurately represent data interactions, ensuring a solid foundation for the database.

Consistency is observed between defined objects and database relationships.

***Define Initial Software Architecture:***

The high-level structure of the software architecture is clear and well-defined.

Interactions between components are clearly outlined, promoting scalability and maintainability.

The initial software architecture aligns with the project objectives.

***Design Initial User Interface Requirements:***

The initial UI design aligns well with the overall project objectives.

The designed UI is user-friendly, considering factors like navigation and layout.

UI requirements align with the defined software architecture, ensuring consistency.

- **Work Items: Planned compared to actually completed**

All the work items planned in this iteration have completed on time.

- **Assessment against Evaluation Criteria Test results**

-All uses cases are defined in a way of aligning project scopes and objects. Also, all of use cases are written in a fully dressed style.

ReserveWell	V1.3
Iteration Plan	Date: 2/12/2023

- Documentation quality is satisfactory, offering clear information for developers.
- Since the test results are successful and the database updates itself accurately in real-time, it can be concluded that database relationships are correct.
- The high-level structure of the software architecture is clear and well-defined. The initial software architecture aligns with the project objectives. In this iteration, an application is developed to meet the requirements of the main use case which is "Make Reservation" and the database has successfully updates itself.
- Since the quality teams have not evaluate our interface yet, it cannot be known whether favorable responses will be observed or not.
- The initial UI design aligns well with the overall project objectives and the designed UI is user-friendly, considering factors like navigation and layout.
- The application successfully passed the most of the prepared test scenarios. One test scenario cannot be passed which is making reservation with multiple group size. The reason for the test is unsuccessful is that the application is not able to update the capacity of the restaurant. This feature will be developed in the next iteration.
- **Other concerns and deviations**
  - Insights gained from user feedback or testing in the previous iteration prompting adjustments in the current plan.
  - Adjustments made to the scope of the iteration due to evolving project priorities
  - Unforeseen technical challenges or roadblocks.
  - Ensuring thorough testing and quality assurance within the time frame.