Norbert Matyas
30432

Smart Restaurant
Supplementary Specification

Version 1.0

Revision History

Date	Version	Description	Author
19/MAR/19	1.0	Initial Requirements Statement	Norbert Matyas

Table of Contents

Supplementary Specification

1. Introduction

The Supplementary Specification captures the system requirements that are not readily captured in the use cases of the use-case model. Such requirements include:

- Legal and regulatory requirements, including application standards.
- Quality attributes of the system to be built, including usability, reliability, performance, and supportability requirements.
- Other requirements such as operating systems and environments, compatibility requirements, and design constraints.

2. Non-functional Requirements

2.1 Availability

The system will have a SLA[1] of 99%. This translates into a yearly downtime of roughly 3 day and 16 hours, or a monthly downtime of 7 hours and 30 minutes. In this time there will be performed software updates or bug fixes, if necessary.

2.2 Performance

Performance is a key factor for our system. For this reason we can allow a response time of up to 10 seconds for request submissions in the worst case scenario. The average response time, depending on the load of the system, should be 1 second.

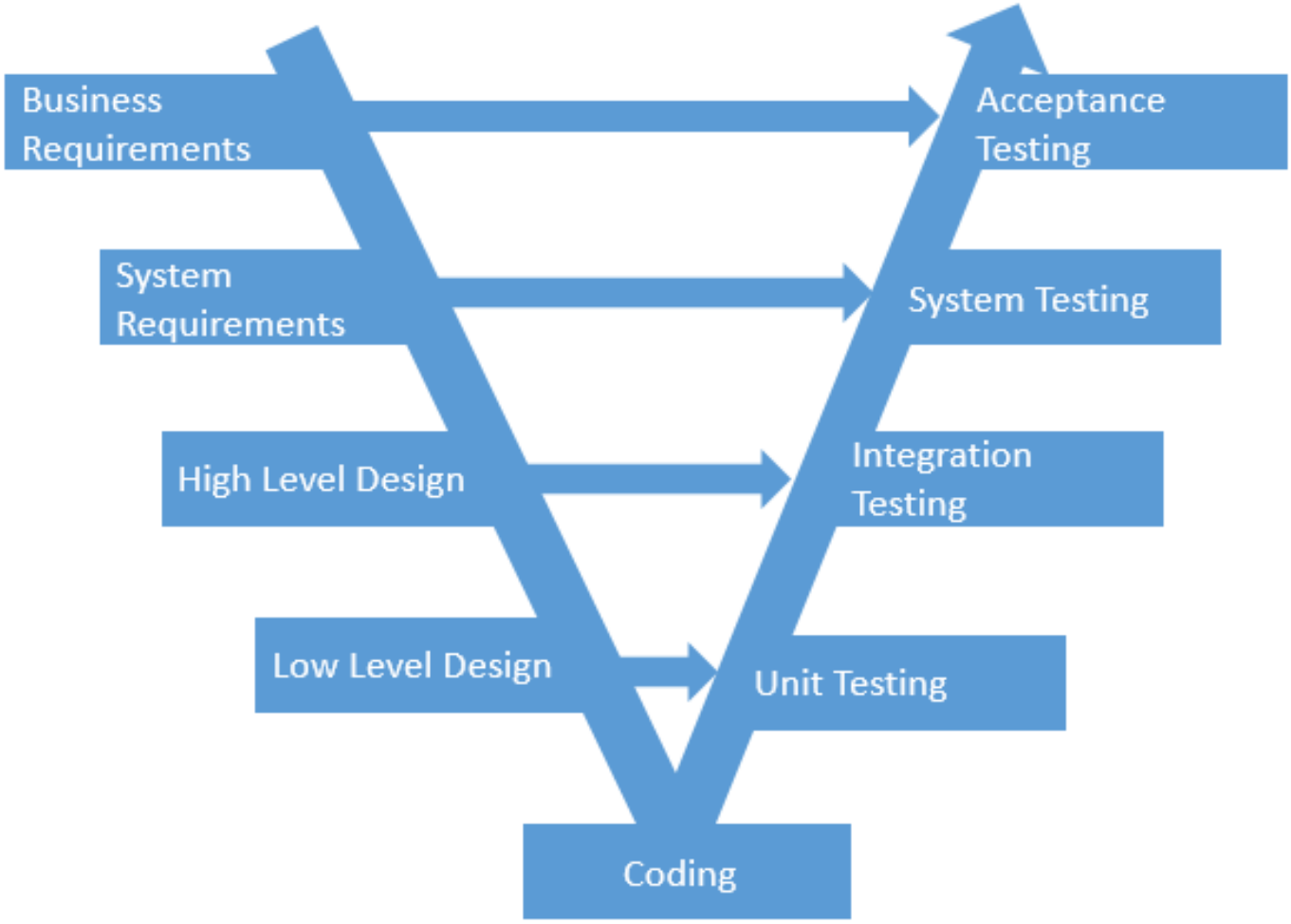
2.3 Security

For this system the security should not be a critical factor, because it does not have any important or sensitive personal information about its users.

2.4 Testability

The business logic of the application must be tested independently from the user interface. We will employ V-Model testing as illustrated in the figure below. We aim to have over 90% test coverage, through unit and integration tests. With respect to manual testing, the system will log all information that is not displayed in the user interface, so that the system is fully observable and testable.

[1] SLA = Service Level Agreement = Availability



2.5 Usability

The user should be able to reach any desired goal in under 30 mouse clicks. Also, at the terminal operations will it prompt a confirmation dialog that describes the consequences of the action.

3. Design Constraints

The system is constrained to use Java 8 as implementation language. The software development process will be the Rational Unified Process (RUP), tailored to fit the team and the project. The conceptual architecture of the system will be a client server as illustrated in the figure below. The required development tools are either Eclipse IDE or IntelliJ IDEA.