Supplementary Specification

for

Tutoria

Version 2.0

Prepared by Nian Xiaodong, Tang Chi Ian,
Wang Junjie, Wang Michelle Yih-chyan

Panacea Inc.

November 12, 2017

Revision History

Name	Date	Reason for Changes	Version
Nian Xiaodong, Tang Chi Ian, Wang Junjie, Wang Michelle Yih-chyan	October 28, 2017	Original	1.0
Nian Xiaodong, Tang Chi Ian, Wang Junjie, Wang Michelle Yih-chyan	November 12, 2017	Refinement	2.0

Table of Contents

Supplementary Specification

1. Introduction

1.1 Purpose

1.1.1 This document serves to capture the system requirements which are not comprehensively captured in the use cases of the use-case model.

1.2 Scope

1.2.1 This document specifies the non-functional and quality attributes of Tutoria, including usability, reliability, performance, and supportability, as well as the design constraints, security and interfaces.

1.3 Definitions, Acronyms, and Abbreviations

1.3.1 Please refers to the project's Glossary document.

1.4 References

1.4.1 Guido van Rossum, Barry Warsaw, Nick Coghlan (2013). PEP 8 -- Style Guide for Python Code [Online]. https://www.python.org/dev/peps/pep-0008/. Assessed on 27 October 2017.

1.5 Overview

1.5.1 The remainder of this document first introduces the non-functional and quality attributes of Tutoria, in the order of usability, reliability, performance, supportability. This is then followed by design constraints, security, online user documentation and help system requirements, interfaces, and finally applicable standards of the project.

2. Usability

2.1 Training for Normal Users

2.1.1 Users should be familiar with the procedures for booking and canceling a session after a 3-minute tutorial.

2.1.2 Users should be familiar with the procedures for managing his/her available tutoring time slots after a 3-minute tutorial

2.1.3 Users should be familiar with the procedures for managing one's finances and payments after a 3-minute tutorial

2.2 Training for Administrators

2.2.1 Administrators should be familiar with the procedures for managing the accounts, including adding, modifying and removing accounts after a 10-minute training.

2.2.2 Administrators should be familiar with the procedures for managing tutorial time slots after a 5-minute training.

2.3 Task Times for Normal users

2.3.1 Users should be able to book a tutoring session within 3 minutes 90% of the time.

2.3.2 Users should be able to cancel a tutoring session within 2 minutes 95% of the time.

2.3.3 Users should be able to specify the available tutoring time slots for a week within 10 minutes 95%

of the time.

2.3.4 Users should be able to add balance to the wallet system within 5 minutes 90% of the time.

2.3.5 Users should be able to withdraw money from the wallet system within 5 minutes 90% of the time.

2.4 Task Times for Administrators

2.4.1 Administrators should be able to add a new account manually within 5 minutes 95% of the time.

2.4.2 Administrators should be able to remove an account manually within 5 minutes 95% of the time.

2.4.3 Administrators should be able to modify an account manually within 8 minutes 90% of the time.

2.4.4 Administrators should be able to add or remove a booking manually within 3 minutes 90% of the time.

3. Reliability

3.1 Availability

3.1.1 The hours of use are from 03:00 to 02:00 (next day) every day in the year.

3.1.2 Minor maintenance can be done outside the hours of use, from 02:00 to 03:00 every day.

3.1.3 The system should be available 99% of the time during the hours of use.

3.2 Failure and Recovery

3.2.1 The mean time between failures should be higher than 24 hours.

3.2.2 The mean time to repair should be less than 6 hours.

3.3 Accuracy

3.3.1 The payment calculation should be accurate 100% of the time.

3.4 Bugs and Defects

3.4.1 The number of minor bugs, which do not cause problems in general operation of the software but may sometimes lead to additional error messages, should be below 10 per thousand lines of code.

3.4.2 The number of significant bugs, which only cause problems in operation in a localized number of users and could lead to long term operational problems, should be below 1 per thousand lines of code.

3.4.3 The number of critical bugs, which cause significant and catastrophic consequences including the complete loss of data and complete shut-down of the system, should be below 0.1 per thousand lines of code

4. Performance

4.1 Response time

4.1.1 The average time for confirming a booking request for a tutoring session should be 5 seconds, and 2 minutes at maximum.

4.1.2 The average time for confirming a cancelation request for a tutoring session should be 5 seconds, and 2 minutes at maximum.

4.1.3 The average time for confirming an update on the tutoring schedule for a week should be 7 seconds, and 3 minutes at maximum.

4.2 Throughput

4.2.1 The number of booking and cancelation requests handled each second should be higher than 5.

4.3 Capacity

4.3.1 The number of users that the system can accommodate should be higher than 1000.

4.3.2 The number of transaction that the system can accommodate should be higher than 1 million.

4.4 Degraded Modes

4.4.1 When localized significant errors occurred, the performance of the system is degraded, but the remaining parts of the system should maintain at least 50% of normal performance

4.5 Resources

4.5.1 The amount of disk memory required per 1000 bookings should be lower than 10 megabytes.

4.5.2 The amount of data between the server and the user computer should be less than 500 kilobytes per booking.

5. Supportability

5.1 Coding standards and naming conventions

5.1.1 The project is developed mainly using Python, following the Style Guide for Python Code (2013) [1].

5.2 Maintenance Access

5.2.1 The administrator accounts are granted access to most components of the website, and can be used for basic maintenance purposes

6. Design Constraints

6.1 Languages

6.1.1 The project is mainly developed using Python.

6.1.2 The web components are implemented in HTML.

6.1.3 The database components use SQL as the query language.

6.2 Development Tools

6.2.1 The project is developed using the Django Web Development Framework.

7. Security

7.1 Personal data

7.1.1 Personal data include the name, email, phone number, and any other data which could be used to identify the identity of the user.

7.1.2 All the user data are released to other users only with their consents, following data privacy practices.

7.1.3 These data are protected and not accessible to other normal users under normal circumstances.

7.1.4 These data can be retrieved by the administrators for internal use only, and such actions will be recorded and monitored.

7.2 Payment data

7.2.1 The external sources of funding from the user (including their bank accounts, credit card details), are carefully protected and encrypted during the transaction, and disposed afterwards.

7.2.2 The balance in the user account is carefully monitored to prevent any theft or fraud.

8. Online User Documentation and Help System Requirements

8.1 Quick-start walkthroughs

8.1.1 Each newly registered user is guided through different parts of system at the first time of use after the registration.

8.1.2 Three major parts of the system must be introduced to the user through these walkthroughs, including ‘booking and canceling a tutorial session’, ‘managing the available time slots for tutors’ and ‘manage your own wallet’.

8.2 Help page

8.2.1 Other than the quick-start walkthroughs, users should be able to find other explanatory materials through a text based help page. Also, contact methods to the administrators will be found on this page.

9. Interfaces

9.1 User Interfaces

9.1.1 All interaction with Tutoria will occur via a web-based interface.

9.1.2 Tutoria web platform will be accessed through a secure user interface requiring user defined username and password.

9.1.3 The layout of the web interface will be responsive for PC view and mobile view.

9.1.4 Major user interfaces include the registration page, the login page, the schedule page, the booking page, the profile page, the wallet page.

9.2 Hardware Interface

9.2.1 Tutoria will interact only with the provided web server, database server and client end devices.

9.3 Software Interfaces

9.3.1 Tutoria will utilize with Django server to deliver web contents to clients.

9.3.2 Tutoria will interact with SQLite for the database operations.

9.3.3 Tutoria will apply jQuery and Bootstrap for interaction with front end user.

9.4 Communication Interface

9.4.1 Tutoria will be connected to the Internet.

10. Applicable Standards

Server software should meet the requirements of standard Python development. Front-end documents should meet the W3C Recommendation of Web Design and Applications, and additionally compatible with Chrome preferentially.