



Software Requirements Documents for the Date Planner App

Date: May 05, 2020

Prepared By:

Team Name	Byte Our Dust	
Team Members	Alejandro Penaloza Nimra Sami Arturo Serdan	a_penalo@uncg.edu n_sami@uncg.edu aaserdan@uncg.edu
Honor Code Professor Quigley	The group Byte Our Dust has abided by the UNCG <i>Academic Integrity Policy</i> on this project.	University of North Carolina at Greensboro

Table of Contents

1.	Introduction	
1.1.	Purpose	3
1.2.	Document Conventions	3
1.3.	Intended Audience	3
1.4.	Definitions/Jargon	4
1.5.	Project Scope	4
1.6.	Technical Challenges	4
1.7.	References	4
2.	Overall Description	
2.1.	Product Features	4
2.2.	User Characteristics	5
2.3.	Operating Environment	5
2.4.	Design and Implementation Constraints	5
2.5.	Assumptions and Dependencies	5
3.	Functional Requirements	
3.1.	Primary	5
3.2.	Secondary	6
4.	Technical Requirements	
4.1.	Operating Systems/Compatibility	6
4.2.	Interface Requirements	6
4.2.1.	User Interface	6
4.2.2.	Hardware Interface	7
4.2.3.	Software Interface	7
4.2.4.	Communications Interface	7
5.	Nonfunctional Requirements	
5.1.	Performance Requirements	7
5.2.	Safety/Recovery Requirements	7
5.3.	Security Requirements	7
5.4.	Policy Requirements	7
5.5.	Software Quality Attributes	7
5.5.1.	Availability	7
5.5.2.	Correctness	8
5.5.3.	Maintainability	8
5.5.4.	Reusability	8
5.5.5.	Portability	8
5.6.	Process Requirements	8

5.6.1.	Development Process Used	8
5.6.2.	Time Constraints	8
5.6.3.	Cost and Delivery Date	8

1. Introduction

1.1. Purpose

The System Requirements Document is to give a comprehensive summary of the requirements of the Date Planner App. This document will demonstrate the scope and comprehensive explanation of the development of the system. It will also explain system restrictions, interface, and intercommunications with other external applications. This application is developed by the group Byte Our Dust, as a project for a Software Engineering course at the University of North Carolina at Greensboro.

1.2. Document Conventions

This document is constructed as follows:

- Section one is the introduction, this section contains all the basic information related to the application.
- Section two is the overall description, this section contains features and characteristics of the application.
- Section three is the functional requirements, this section is there to define the functions of the application.
- Section four is the technical requirements, this section contains information related operating systems and interface requirements.
- Lastly, section five is the nonfunctional requirements, this section contains important requirements related to the application.

1.3. Intended Audience

This document is developed by all the group members of Byte Our Dust and it used to demonstrate knowledge of Software Engineering. This is to show the applied knowledge in all the different principles and terminologies learned through the course of Software Engineering. This document is intended for the person inquiring the use of the Date Planner application.

1.4. Definitions/Jargon

Term	Definition
API	Application Programming Interface
UUID	Universally Unique Identifier

1.5. Project Scope

The Date Planner App is a date planning application that helps couples plan a perfect date for them, it eases the struggle of planning a date by providing the users with options of different restaurants around the users based on their given data. This application should be free of cost and will be compatible with operating systems with a java virtual machine. This software will need an internet connection to fetch the results from the Yelp API and weather by using the weather API and then will display results.

1.6. Technical Challenges

To start the database, it was hard to find a way to read from the text file starting from the last element added instead of the other way around. Also learning how to use the UUID in order to create a search on the file and on the order. It was tough to figure out how to retrieve the data from the API's in a way that made it easy to use throughout the program

1.7. References

“Academic Integrity Policy - Revised 5-8-2017.Pdf. ” *UNCG*,
drive.google.com/file/d/0B0rFGGhJvbDHUExSZmFFaWFmb00/view.

2. Overall Description

2.1. Product Features

With the help of this application, the users will be able to plan a perfect date for themselves. Two users will be able to get together and input what they are in the mood for when planning a date, and then the application gives back a list of all of the options of different restaurants. The results will be based on the criteria for the user's inputs. The result of the search will be viewed in a list view in two different columns, it will have as many results as the users asked for. The application will display results in two different columns with the user's names and each listview will have the name of the restaurant, restaurant address, restaurant ratings,

restaurant price range and also their number. It will also provide users with the weather in the area (Temperature in Fahrenheit and a weather description).

2.2. User Characteristics

Two types of users interact with this software, users of the application, and the administrators. Both types of users would have a different use of the software so each will have their requirements. The users of the application can use this application to plan a perfect date. This means the user will have to fill in the requirements based on their location and preferences and wait for their results. The administrators will not use the application themselves but instead will use the software to manage the overall system so there is no false information within it and to allow for scalable changes to add more features in the software as needed.

2.3. Operating Environment

This software must operate with a JAVA IDE operating system because this software is developed in Java programming language. Java IDEs can run on almost every major operating system.

2.4. Design and Implementation Constraints

When designing the application it was originally planned for the users to simply enter the city they lived in, in order for the program to grab both the restaurant and weather information. However, there were no API's that fulfill these requests with just the city name, as the weather API requires a zip code. So when implementing the design developers had to require both a city name and zip code.

2.5. Assumptions and Dependencies

An assumption about this product is that it will be used on a java virtual machine. If your laptop does not have a JAVA IDE operating system, then the user will not be able to use this software. Some dependencies include connections to two API's that require an internet connection.

3. Functional Requirements

3.1. Primary

The Primary Functional requirements of the date planner app are to assist the users in the process of planning a perfect date for themselves. It accomplishes this through the implementation of a user-friendly interface, that asks simply for one's city, zip code, and food cravings.

3.2. Secondary

The Secondary Functional requirements of the Date Planner App are to assist the users with the option of starting over. If users are not happy with their results, the application allows them to start over.

4. Technical Requirements

4.1. Operating Systems/Compatibility

This application will be compatible with operating systems with a java virtual machine. JAVA IDE is really important to operate this software.

4.2. Interface Requirements

4.2.1. User Interface

A first-time user of this application should see the homepage whenever the user runs the application, as shown in Figure 1. Then the users should be able to plan a date by clicking the "Plan a date" button. Then it should take the users to the plan page, as shown in Figure 2. Here is where the users input their city, zip code, their names and they choose the cuisine and their search limit. Then the user clicks for results and then results will appear in the list view in two different columns, as shown in Figure 3. Then from here, users can read about the application in about us tab, as shown in Figure 4 and they can also contact the administrators of the application through the contact us page, as shown in Figure 5.

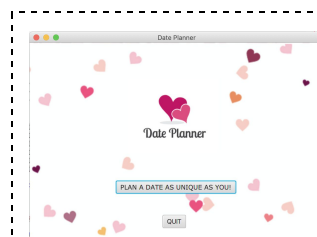


Figure 1: Home Page

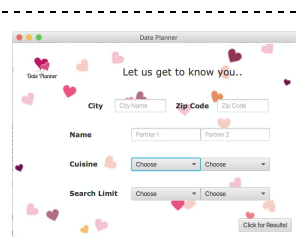


Figure 2: Plan Page

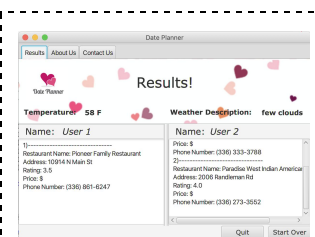


Figure 3: Results Page

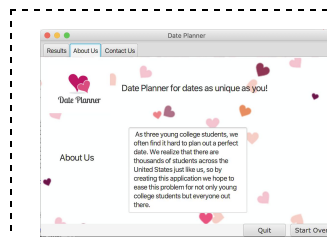


Figure 4: About Us Page

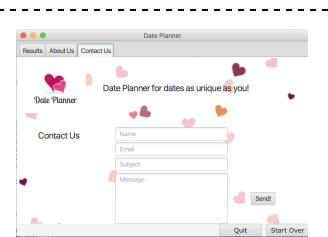


Figure 5: Contact Us Page

4.2.2. Hardware Interface

Since desktop does not have any designated hardware therefore it does not have any direct hardware interfaces.

4.2.3. Software Interface

This application uses JAVA, so a JAVA IDE would be required.

4.2.4. Communications Interface

This application communicates with two different APIs, to fetch the needed information. It communicates with Yelp API to get restaurant suggestions and their detailed information. It also communicates with the Weather API to get the weather and the weather description.

5. Nonfunctional Requirements

5.1. Performance Requirements

The Date Planner App requires a system with the operating system of Windows 7, Windows 8, Windows 10, Mac OS X, and or Linux.

5.2. Safety/Recovery Requirements

Information communication should be securely transferred to the server without any changes to the code and if it breaks down it will require a full reinstallation to recover all the issues.

5.3. Security Requirements

The Date Planner App does not have any security requirements and therefore any type of user can use it without any additional perquisites.

5.4. Policy Requirements

Policy requirements to follow in order to use this application is that users will agree on giving information such as their locations and their names.

5.5. Software Quality Attributes

5.5.1. Availability

At the moment this application is only available for the team members of Byte Our Dust, but anyone could use this application if the user has installed it on their machine.

5.5.2. Correctness

The accuracy of the application in its choices of restaurants is very reliable. It uses a weather api and the Yelp api which require the users zip code AND city to ensure proper and accurate results.

5.5.3. Maintainability

This application's structure, design, implementation, and documentation will be maintained by the developers to keep everything up to date.

5.5.4. Reusability

This application is designed in a way that it could be reused for other purposes.

5.5.5. Portability

A user should be able to install and run this application on a desktop which meets all of the requirements stated in this document.

5.6. Process Requirements

5.6.1. Development Process Used

This application was developed by team Byte Our Dust using the Agile method.

5.6.2. Time Constraints

This application will be developed in approximately three to four months. The initial start date was January 21st, 2020 and the expected date to finish is May 5th, 2020.

5.6.3. Cost and Delivery Date

This application is free of cost and the delivery date is May 5th, 2020.