Software Design Description

for

TODO: Task Management System

CS 471

Draft 0.0.4

Prepared by:
Blake Eggemeyer

November 20, 2011

Contents

1 Introduction

1.1 Overview

1.2 Stakeholders

The stakeholder in the design is also the client.

1.3

1.4 Definitions

1.4.0.1 Immediately: Immediately refers to actions that will begin as soon as the user has given the input for the action to occur. This applies to the reordering of the list when new input is given. The action will take a measurable, non-zero amount of time.

1.4.0.2 Should: Requirements with this marker are desired, but not crucial, and will be a part of the final deliverable contingent on time and progress.

1.4.0.3 TBD: Acronym for To Be Determined. This is used in this document to signify that the information necessary for a part of this document is “To Be Determined”.

1.4.0.4 TODO: Working name of the project.

1.4.0.5 User: The person, or persons, who operate or interact directly with the product.

1.4.0.6 Will: Requirements with this marker are guaranteed to be in the final delivered product.

1.5 References

The `1998 - IEEE Standard for Information Technology - Systems Design - Software Design Descriptions` was referenced to produce this document.

1.6 Revision tracking

0.0.1	Nov 11	Empty document created.
0.0.2	Nov 17	Framework added.
0.0.3	Nov 18	Framework extended.
0.0.4	Nov 19	Design.

2 Design Considerations

2.1 Programming Languages

TODO will be implemented in C++ due to the programmers experience with that language.

2.2 Project Management

This project will use `Git` version control in conjunction with `GitHub` to keep track of changes.

3 Data Storage

3.1

3.2 Data Dictionary

3.2.1 Task

3.2.1.1 Unique ID: Non user editable `int`. This is the index value used for internal reference to the task.

3.2.1.2 Name: Data type `string`.

3.2.1.3 Description: Data type `string`.

3.2.1.4 Project: Data type `string`.

3.2.1.5 Due date: Input as three integers. Stored as Unix epoch time `int`.

3.2.1.6 Time estimate: Input as three integers. Stored as Unix epoch time `int`.

3.2.1.7 Elapsed time: Calculated based on ‘Working on top item in list’ function. Stored as Unix epoch time `int`.

3.2.1.8 Priority: Integer representing number of tasks from the top.

3.2.1.9 Prerequisites: Integer representing the unique Id of another task.

3.2.2 Appointment

3.2.2.1 Unique ID: Non user editable `int`. This is the index value used for internal reference to the appointment.

3.2.2.2 Name: Data type `string`.

3.2.2.3 Description: Data type `string`.

3.2.2.4 Project: Data type `string`.

3.2.2.5 Date: Input as three integers. Stored as Unix epoch time `int`.

3.2.2.6 Estimated duration: Input as three integers. Stored as Unix epoch time `int`.

3.2.2.7 Time worked: Calculated based on ‘Working on top item in list’ function. Stored as Unix epoch time `int`.

3.2.2.8 Priority: Integer representing number of tasks from the top. This is re-calculated every minute to move the appointment up the active list.

3.2.3 Active list

The active list will be recalculated form the complete list each time a recalculate is requested. This will occur when changes are made to the items in the list and at scheduled intervals.

3.2.4 Inactive list

4 Views

4.1 Viewing Active List

The user will see the list of tasks and appointments to be done in order of priority. These tasks will be those that are

4.2 Viewing All

The user will see the entire list of work done on tasks regardless of completion status.

5 Interface

5.1 Command line

Using a command line interface would allow the TODO software to be used in an open Linux terminal. This interface option is for more limiting for typical users than a GUI.

5.2 Glade2

Glade2 is a user interface design tool that would allow TODO to be implemented in a GUI.

6 Actions

6.1 Command line

6.1.0.1 Working on top item in list: By running `todo top` the user will initiate the accumulation of time worked on the top item in the list. If there are no items in the list, the program will return test to the command shell stating `No items in list`.

6.1.0.2 Mark inactive: By running `todo inactivate [Unique ID]` the specified task will be marked inactive and removed from view.

6.1.0.3 Completed: By running `todo complete` the top task in the list will be marked as inactive and completed, and the time worked will stop accumulating.

6.1.0.4 Done for now: By running `todo stop` the top task in the list will stop accumulating time worked.

6.1.0.5 Changing list order: By running `todo move [Unique ID] [new priority]` the specified task will be moved to the specified priority position in the list, unless the prerequisites for that task are ahead of it. In this case the option will be given to move all prerequisites of this task to the specified position.

6.1.0.6 Modify a user created field: By running `todo modify [Unique ID] [option]` the specified task will be modified, with the given piece of data being changed to the given value.

6.1.0.7 Make item top priority: By running `todo move [Unique ID] 0` he specified task will be moved to the top priority position in the list, unless the prerequisites for that task are ahead of it. In this case the option will be given to move all prerequisites of this task to the specified position.

7 Test Cases

8 Traceability matrix

	3.1.1	3.1.2	3.1.3	3.1.4	3.1.5.1	3.1.5.2	3.1.6.1	3.1.6.2	3.1.6.3	3.1.6.4	3.1.6.5	3.1.7.1	3.2.0.1	3.2.0.2	3.2.0.3	3.2.0.4
3.2.1	✓															
3.2.2		✓														
3.2.3			–													
3.2.4				–												
6.1.0.1							✓									
6.1.0.2								✓								
6.1.0.3									✓							
6.1.0.4										✓						
6.1.0.5						✓					✓					
6.1.0.6					✓											
6.1.0.7						✓										
??																
??																