FunSheets software

Team 2 - Iteration 1

3 January 2013

Table 1: Team

Name	ID Number
Kevin Cameron	9801448
Addison Rodomista	1967568
Dragos Dinulescu	6304826
Adrian Max McCrea	9801448
Ghazal Zamani	1971158
Karim Kaidbey	9654726
Carmelo Fragapane	6298265
Long Wang	9547967
Simone Ovilma	9112510
Nicholas Constantinidis	6330746
Asmaa Alshaibi	9738231

1 System

FunSheets is an electronic spreadsheet program used for storing, organizing and evaluating raw data. FunSheets is designed, written and distributed by Team 2 of COMP354 W/PP.

1.1 Purpose

FunSheets is used to automate data orginization and the calculations of complex expressions involving large amounts of data.

1.2 Context

FunSheets will allow the user to create, open or modify a spread sheet and evaluate the source to produce a new spreadsheet featuring calculated value, evaluated expressions, and sorted information.
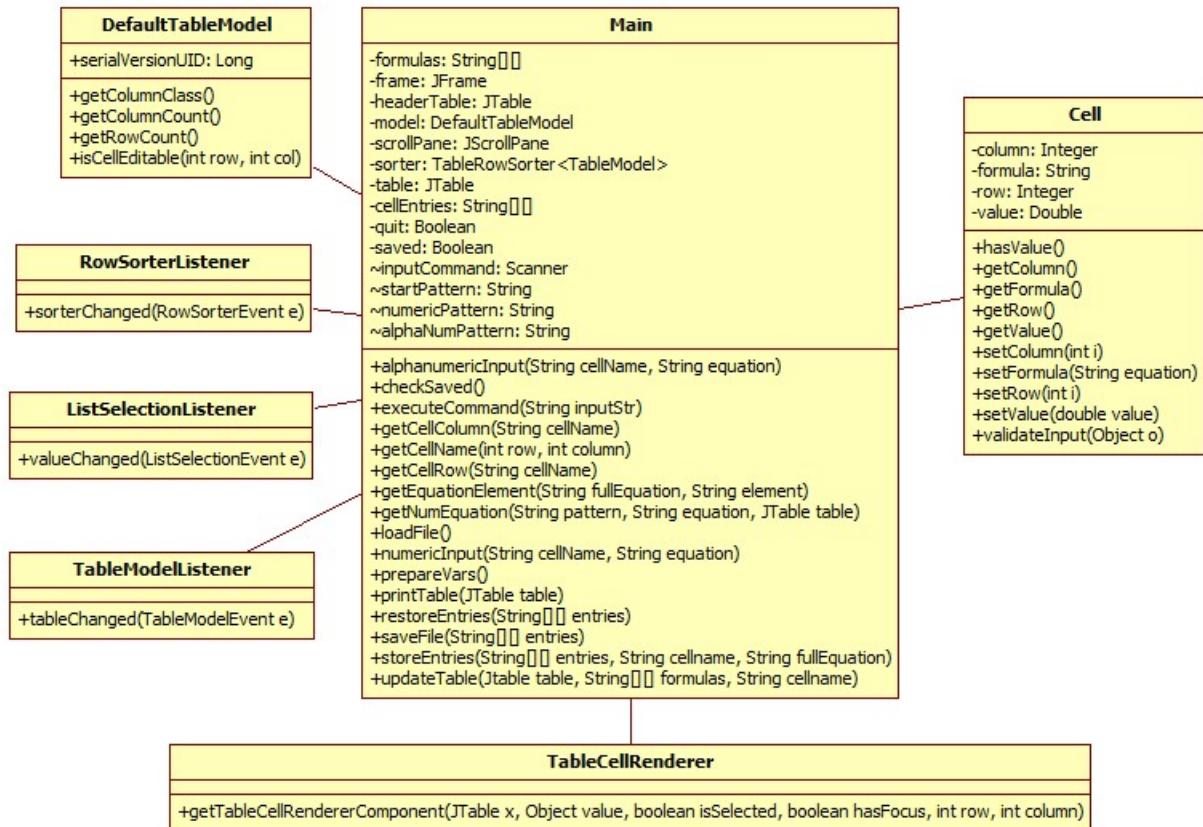
2 Domain Concepts

2.1 Domain Model

Figure 1: Domain Model

2.2 Main Domain concepts of the users

Table 2: Concepts

Concept	Description
Registered Users	A person that has a registered account on funsheets.com can download the software for free and will have access to all future updates
Other Users	All other people can download a 30-day trial

3 Actors

A list of actors (primary/secondary) and use cases are mentioned in this section.

3.1 List of Actors

1. Person as a primary actor.

2. System as a secondary actor.

3.2 Description of Actors

1. Person: The main user of the application. Their role is to input raw data or expressions manually or by opening existing files.

2. System: A software application that provides utilities for spreadsheet analysis and evaluation.

4 Use Cases

4.1 Overview

The Use Case model provides an understanding of the interaction between the user and the software system. It consists of one or more actors and one or more use cases where actors represent particular role and use cases represent actions performed by the system.
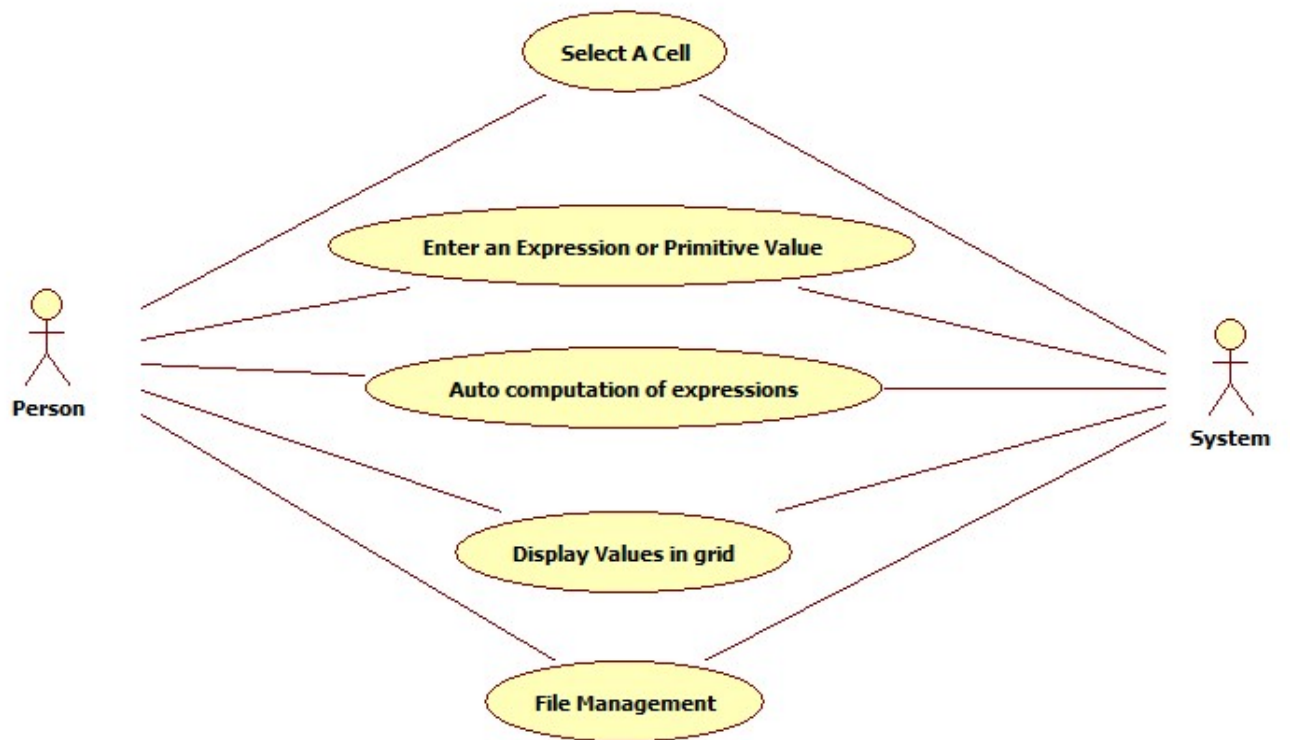
4.2 Use Case Diagram

Figure 2: Use Case Diagram

Use Case 1

Name
Select a Cell

Summary
A user will select a cell that can be used for various operations.

Actors
1. Person
2. System

Precondition
The user has opened the program, and opened or created a spread sheet.

Main Scenario

1. User will click on the desired cell or enter the cell address.

Exceptions
The cell is outside an acceptable range.

Postcondition
Selecting a cell will trigger an event in the system to evaluate the input.

Use Case 2

Name
Enter Formula / Primitive value

Summary
The user can enter a primitive value of type real which must have at least one digit before the decimal point, and if the real number has a decimal part then there is at least one digit after the decimal point. The numbers may be negative. In addition, the user can enter a formula that starts with an equals sign (=) and these formulas are the usual arithmetic expressions.

Actors
1. Person
2. System

Precondition
The user has selected a cell.

Main Scenario

1. Select a Cell

2. Enter primitive data through standard input devices (eg. keyboard, scanner, etc.)

Exceptions
None

Postcondition
Data stored in the selected cell of the spread sheet.

Use Case 3

Name
Automatic Computation

Summary
When a cell contains an expression the system will parse the string and evaluate the elements to produce a value.

Actors
1. System

Precondition
A cell contains an expression.

Main Scenario

1. User opens the program.

2. User enters an expression, or loads a sheet with expressions.

3. User selects Automatic computation.

Exceptions
None

Postcondition
The cell that contains the expression will display the results of the evaluation.

Use Case 4

Name
Display On Screen

Summary
Output the spreadsheet as a grid of cells with values and evaluated expressions.

Actors
1. System

Precondition
Program is open
A spreadsheet has been created or loaded.

Main Scenario

1. User opens the program.

2. User enters an expression or loads a sheet with expressions.

3. System will display the results in a grid of cells.

Exceptions
None

Postcondition
A GUI representation of the grid is displayed with evalutated cells in a row/column grid.

Use Case 5

Name
File management

Summary
Allows the user to load or save an existing spreadsheet to disk.

Actors
1. System
2. User

Precondition
Program is open.

Main Scenario

1. User opens the program.

2. User loads an existing file.

3. User can make modifications to the spreadsheet.

4. System will evaluate the results.

5. User saves the changes to disk.

Exceptions
None

Postcondition
A GUI representation of the grid is displayed with evalutated cells in a row/column grid.

5   Non-Functional Constraints

ID	Ver	Requirement	Source	Rationale	Status	Traces to use case
No.1	1	User must be registered	IT manager	access information about the users	implemented	Register, login, download

6   Data Dictionary

7   References

More information about .csv http://en.wikipedia.org/wiki/.csv
More information about jTable http://docs.oracle.com/javase/6/docs/api/javax/swing/JTable.html

A   Description of File Format: Input

The application accepts a .CSV file (Comma separated value). A CSV file consists of any number of records, separated by line breaks of some kind. The line break represent the start of a new row of data. Each Row consists of fields, separated by some other character or string, most commonly a literal comma (,) or tab. Usually, all records have an identical sequence of fields.

B   Description of File Format: Output

Display Output:
The display output is a grid of cells that are arranged into rows and columns. Each cell that contains primitive data are displayed without modification to their contents. Cells that contain expressions are displayed with their expressions evaluated into usable data.

File Output:
When a spreadsheet is saved each cell is converted to a field and exported to a CSV file. Cells that contain primitive values will be exported without modification. Cells that contain expressions will be exported as a string representation of the expression.