

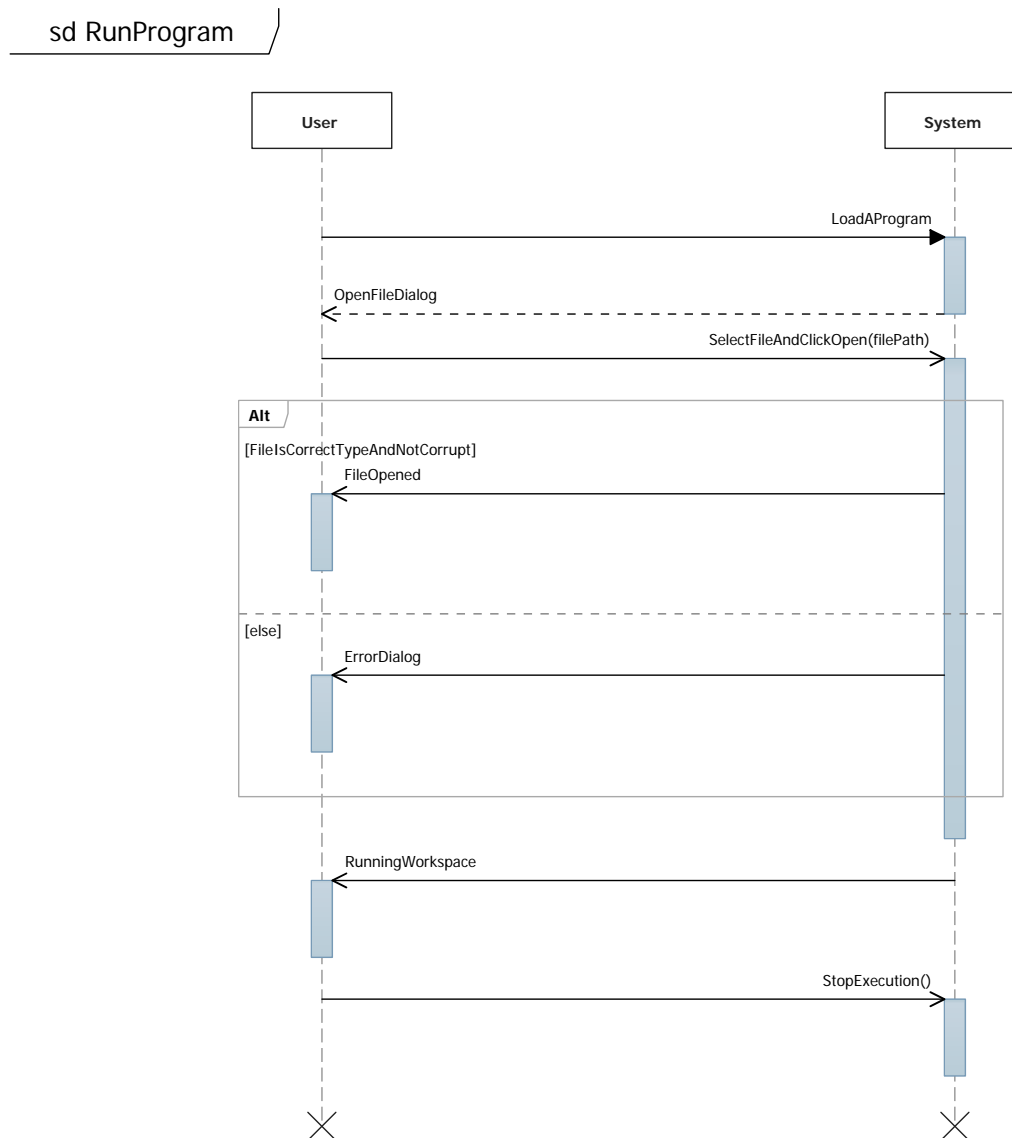
Siftables Emulator  
*Singularity Software*

January 13, 2012

Alex Mullans  
Ethan Veatch  
Eric Vernon  
Kurtis Zimmerman

# 1 System Sequence Diagrams

One system sequence diagram was created to describe the action of loading and beginning the execution of an application in the workspace. No further system sequence diagrams were deemed necessary because the rest of the user-system interactions are trivial.



## 2 Operation Contracts

### OC1: SelectFileAndClickOpen

<b>Operation</b>	SelectFileAndClickOpen(filePath : String)
<b>Cross-references</b>	UC1: Load program, UC2: Reload program
<b>Preconditions</b>	The OpenFileDialog is open.
<b>Post-conditions</b>	The file name was parsed. The emulator opened the file.

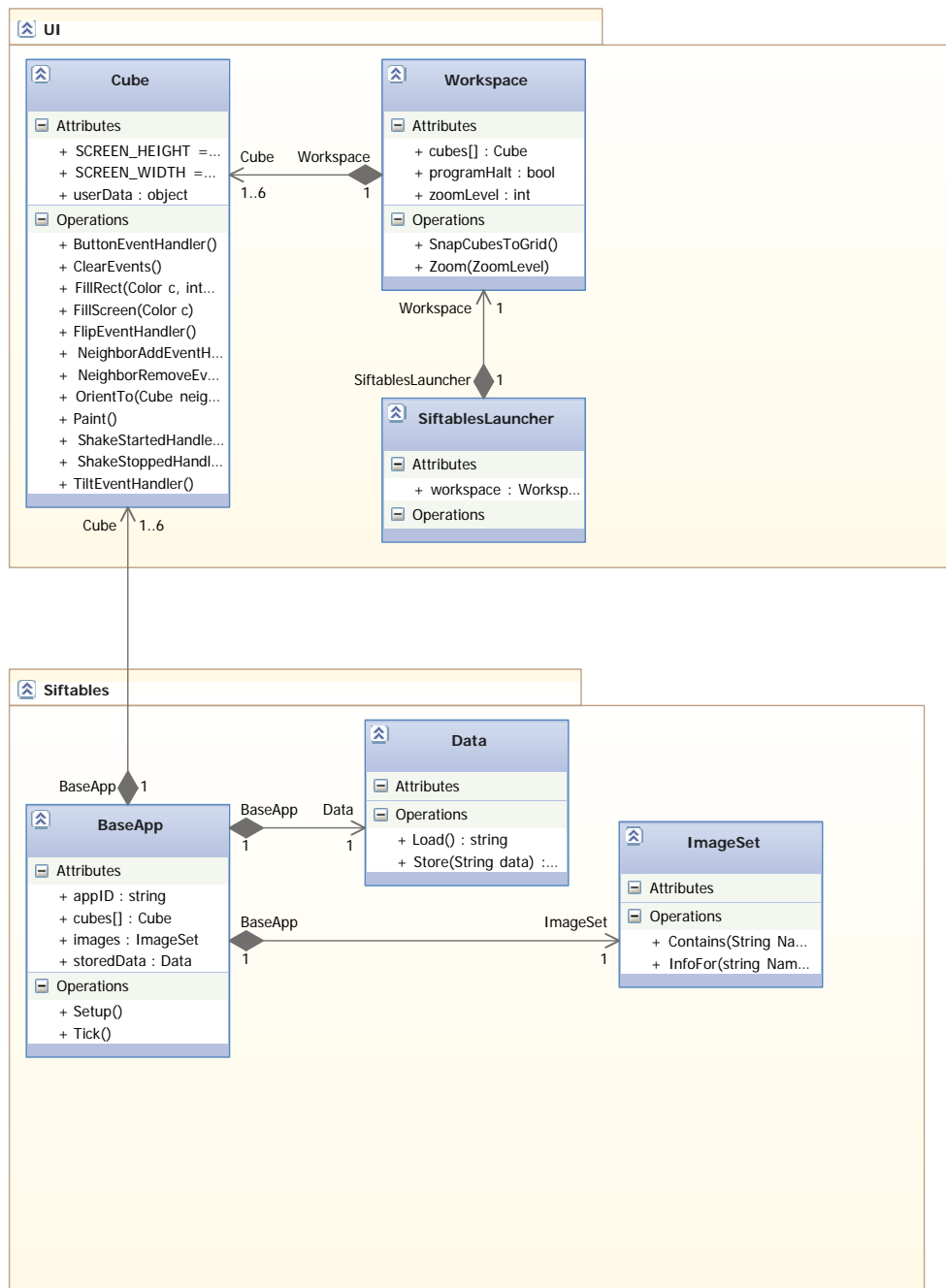
### OC2: ZoomSliderChanged

<b>Operation</b>	ZoomSliderChanged(zoomLevel : int)
<b>Cross-references</b>	UC3: Zoom screen
<b>Preconditions</b>	There is an application running in the workspace.
<b>Post-conditions</b>	The workspace canvas has been magnified appropriately. The workspace zoomLevel attribute was updated.

Additional operation contracts for the remaining use cases were not pursued because of their trivial nature. The basic format of this operation contract — user changes UI element, UI adjusts accordingly, and program updates relevant attributes — applies to the other use cases.

### 3 Design Class Diagram

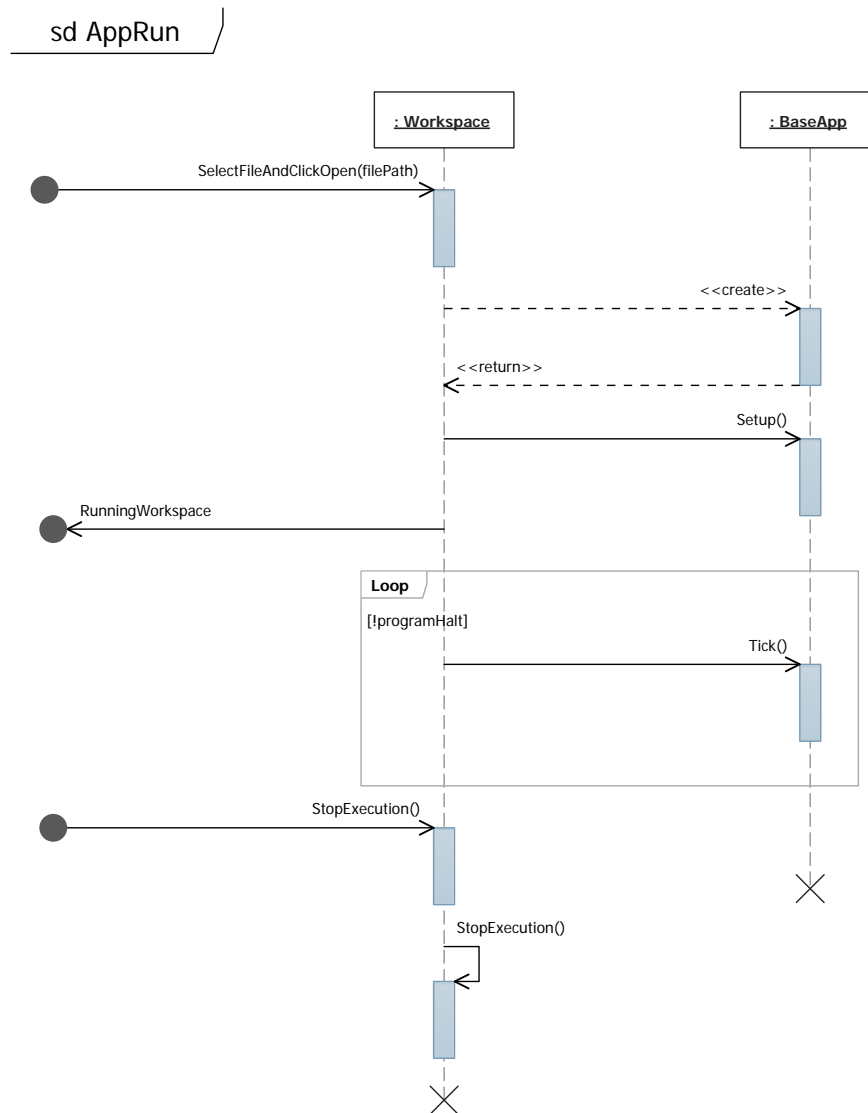
cd ClassDiagram



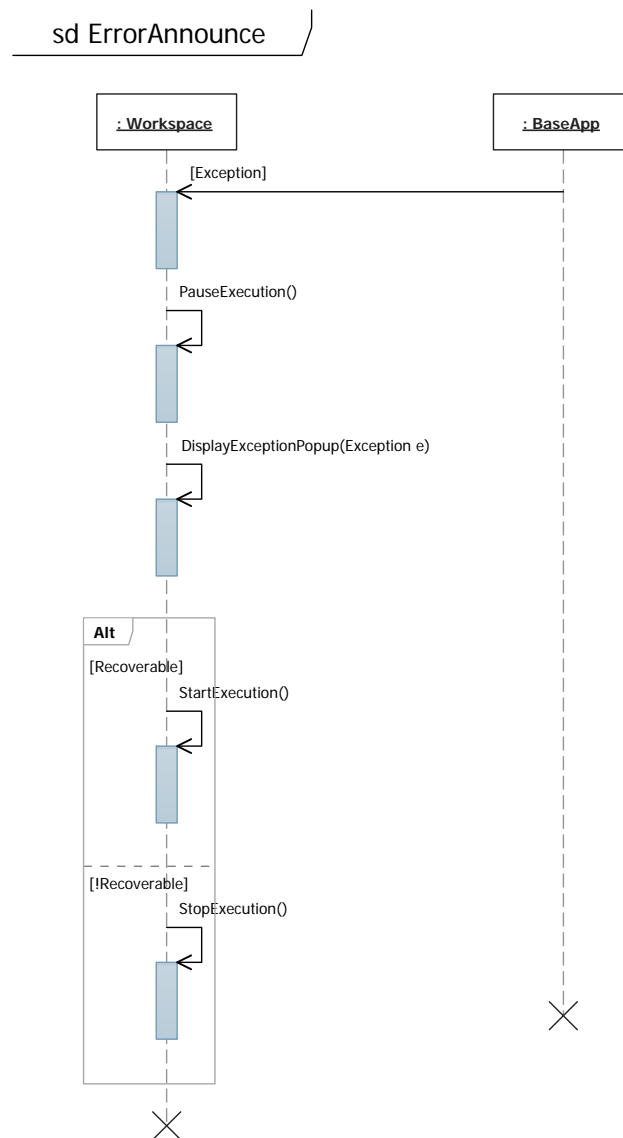
At this stage in the design of the emulator, there are a few basic entities designed to show the relationship between the UI objects and the domain layer. Cube in the UI layer is responsible for rendering the graphics on a screen; applications, which must extend BaseApp, are responsible for sending messages to the Cubes to make changes.

## 4 Sequence Diagrams

The following sequence diagrams, despite appearing fairly trivial, depict the general functionality of the system in a level of detail sufficient to begin implementation.



The AppRun SD shows the pattern followed by about 75 percent of the system commands and is thus crucial for moving into product development.



The ErrorAnnounce SD shows the error handling lifecycle pattern we expect to use throughout the project.