101 Solutions

Iteration 2

101 Solutions

August 11, 2013

Contents

1 Introduction

The purpose of this document is to keep track of the ongoing development progress of the software application manager AppMan. In each iteration additional content will be added which will result in this document being an up to date and relaible method to view progress and current development plans and ideas. Content will include milestones reached, current goals and the current functionality of the software.

2 Milestones reached

- A cleaner GUI implementation
- Server running on the Master - AppMan
- Client connection to the master through network
- A reliable socket connection between the server and client applications, which will serve as a communication pathway
- Build add to the master computer
- View builds on the master
- Creating a database that can store information related to builds kept on the master and slaves

3 Immediate Goals

Our next main goals for the project:

3.1 Build transference

- Goal: The ability to copy the builds to the clients(Slave computers) and also to compress the files that will be copied

3.2 Build Information Comparison

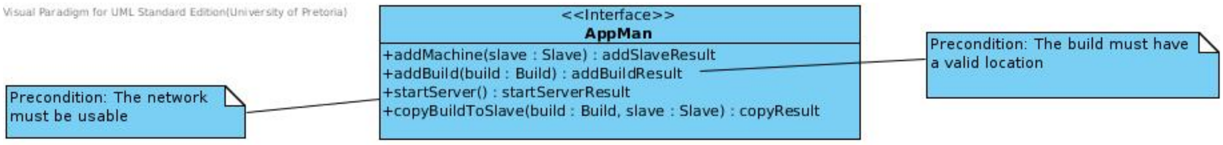
- Goal: The ability to synchronize two builds on master and slave computers(i.e. if a change have been made to a file)

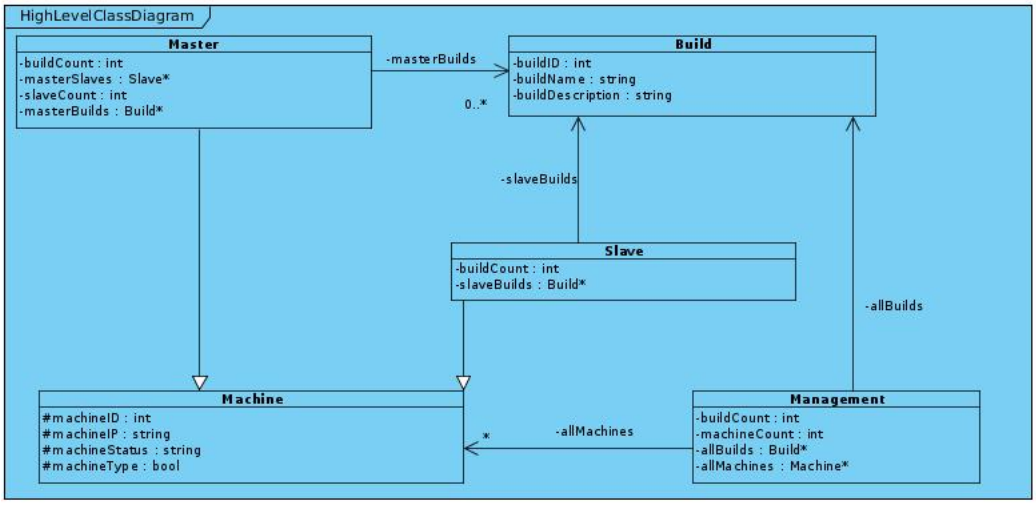
3.3 Slave Build Display

- Goal: The ability to show the builds on slave machine

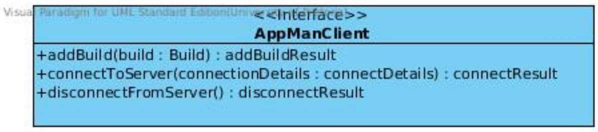
4 System Design

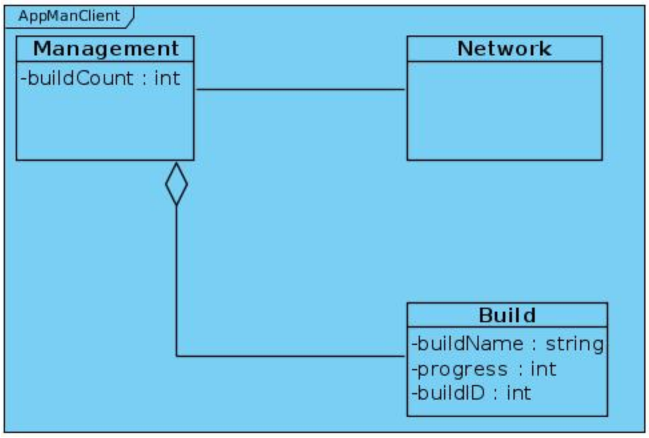
4.1 AppMan


Visual Paradigm for UML Standard Edition(University of Pretoria)
<<Interface>>
AppMan
+addMachine(slave : Slave) : addSlaveResult
+addBuild(build : Build) : addBuildResult
+startServer() : startServerResult
+copyBuildToSlave(build : Build, slave : Slave) : copyResult
Precondition: The network must be usable
Precondition: The build must have a valid location



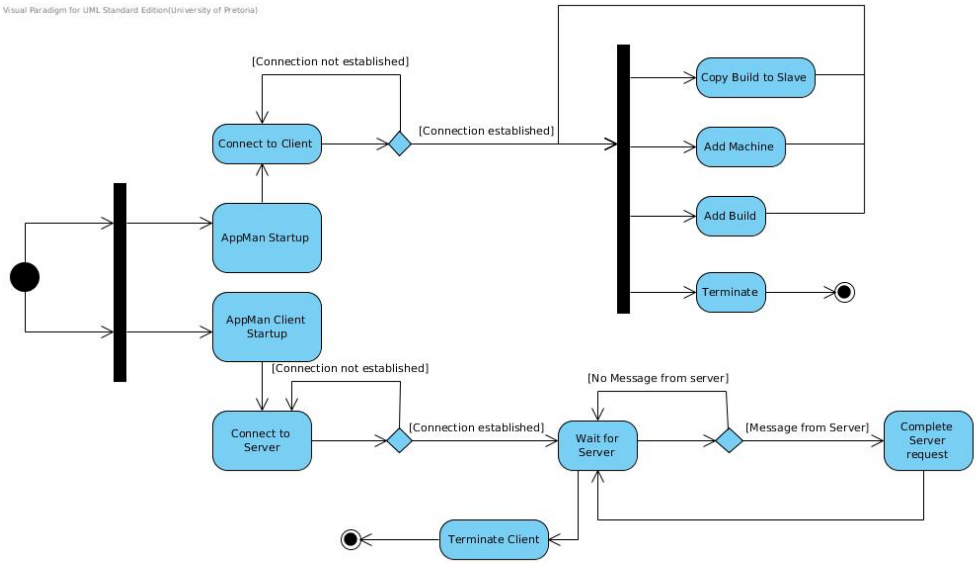
4.2 AppManClient


Visual Paradigm for UML Standard Edition(University of Pretoria)
<<Interface>>
AppManClient
+addBuild(build : Build) : addBuildResult
+connectToServer(connectionDetails : connectDetails) : connectResult
+disconnectFromServer() : disconnectResult



5 Processes

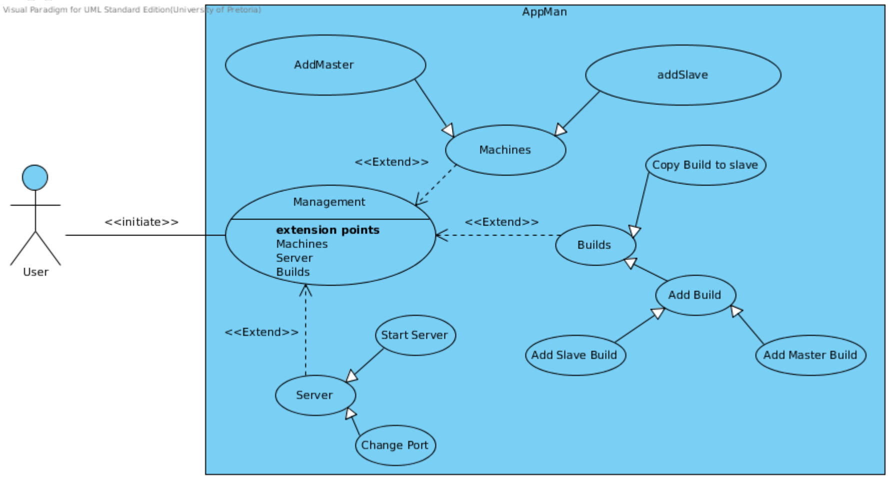
5.1 ActivityDiagram



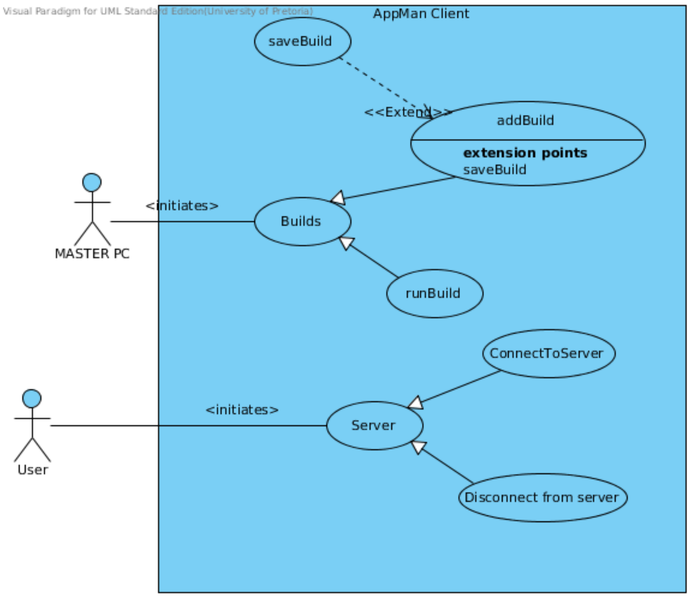
6 Functional Requirements

6.1 Use Case Diagrams

- AppMan



- AppManClient



6.2 Builds

- Builds must be addable to the master
- Builds must be updated on the slave computers once a change have been made

- Only changed files must be sent to the slave computers once a change has been made
- Easy to manage the builds that are on the master computer
- Builds must be able to have the following
  - Build Number
  - Directory
  - Name
  - Description

6.3 Copying

- Copying must be done over the network
- The copy of multiple files must be compressed to reduce the time it takes to copy a file
- Progress of copy must be shown on master computer while copying takes place

6.4 User Interface

- The user interface must be easy to use
- The user interface must make use of drag and drop capabilities to promote usability

6.5 Server & Client Applications

- The Applications must start on computer startup in a minimized form(e.g. in the tray)
- The Master machine server must start on application start
- Slave machines must connect to the master computer via the network once the application starts up
- Slave machines must prompt connection details once and then connect automatically afterwards
- The user interface must make use of drag and drop capabilities to promote usability

7 Glossary

- Build - An application build version that could potentially be distributed to slave computers
- Slave - A computer that will be controlled via a master computer
- Application builds will be sent to this computer
- Master - A computer that will control Slaves across a network
- Server - A machine waiting on the network for connections from other machines
- GUI - Graphical User Interface with which a user can control an application
- Project - This project. The distributed application manager
- Application Configuration - Environment variables that are specified when running an application