**Software Requirements Specification**

**ANNEX MANAGEMENT SYSTEM**

**Problems-**

1. Waiting the students in annex.
2. What times to empty system, students don’t know .
3. Very crowed in annex for getting system.

**Purpose -**

This document provides all of the requirements for the annex management system. We make software for removing of the above problems.

**Scope-**

When students login in system then students will insert burst time, so that other students can know about which time will be empty system. So scope of the student will be valid burst time.

**References-**

1. HTML
2. PHP
3. SCRIPT
4. MYSQL

**Overview-**

This document specifies SRS for annex management system which is store the data about students and system counters. When students will insert username, password and also will insert burst time, system number. When no one system is empty then students can know how will be empty system. And no crowed In annex due to applied of this software.

**Annex\_Management\_System**

**Report on Configuration DefaultConfig**

Packages

com

Language: C++

Nested Packages:

Annex\_management\_system

Globals:

Annex\_management\_system

Language: C++

Object Model Diagrams:

**Annex\_Management\_System**

****

**Overridden Properties**

Subjects:

**Format**

Metaclasses:

**Class**

Properties:

Line.LineWidth: 1

Font.FontColor: 0,0,0

Fill.FillColor: 255,255,255

Font.Font: Arial

DefaultSize: 0,34,84,148

Line.LineStyle: 0

Font.Size: 10

Line.LineColor: 121,122,0

**Object**

Properties:

Font.Underline@Child.NameCompartment@Name: 1

Line.LineWidth: 1

Font.FontColor: 0,0,0

Fill.FillColor: 255,255,255

Font.Font: Arial

DefaultSize: 0,34,84,148

Line.LineStyle: 0

Font.Size: 10

Line.LineColor: 121,122,0

Sequence Diagrams:

**sequencediagram\_15**

****

**Overridden Properties**

Subjects:

**Format**

Metaclasses:

**InstanceLine**

Properties:

Line.LineWidth: 0

Font.FontColor: 0,0,0

Fill.FillColor: 255,255,255

Font.Font: Arial

DefaultSize: 0,0,96,437

Font.Size: 10

Line.LineColor: 121,122,0

**Message**

Properties:

Line.LineWidth: 1

Font.FontColor: 0,0,128

Font.Font: Arial

Font.Size: 10

Line.LineColor: 0,112,0

**EnvironmentLine**

Properties:

Line.LineWidth: 1

Font.FontColor: 0,0,0

Fill.FillColor: 255,255,255

Font.Font: Arial

DefaultSize: 0,0,96,437

Font.Size: 10

Line.LineColor: 0,0,0

**SequenceDiagram**

Metaclasses:

**General**

Properties:

ClassCentricMode: True

RealizeMessages: True

CleanupRealized: True

Use Case Diagrams:

**Annex\_Management\_system**

****

**Overridden Properties**

Subjects:

**Format**

Metaclasses:

**Actor**

Properties:

Line.LineWidth: 1

Font.FontColor: 0,0,128

Fill.FillColor: 255,255,255

Font.Font: Arial

DefaultSize: 0,26,84,168

Font.Size: 10

Line.LineColor: 111,0,107

**UseCase**

Properties:

Line.LineWidth: 1

Font.FontColor: 0,0,0

Fill.FillColor: 255,255,255

Font.Font: Arial

DefaultSize: 0,21,129,92

Font.Size: 10

Line.LineColor: 111,0,107

**System\_Border**

Properties:

Line.LineWidth: 1

Font.FontColor: 0,0,0

Fill.FillColor: 255,255,255

Font.Font: Arial

DefaultSize: 0,0,228,240

Font.Size: 10

Line.LineColor: 111,0,107

**Inheritance**

Properties:

Line.LineWidth: 1

Font.FontColor: 0,0,128

Font.Font: Arial

Font.Size: 10

Line.LineColor: 0,16,230

**Flow**

Properties:

Line.LineWidth: 0

Font.FontColor: 0,0,128

Font.Font: Arial

Line.LineStyle: 1

Font.Size: 10

Line.LineColor: 0,112,0

**Association**

Properties:

Line.LineWidth: 1

Font.FontColor: 0,0,128

Font.Font: Arial

Font.Size: 10

Line.LineColor: 221,0,0

Events:

**evclose**

**evKeyprocess**

**evopen**

**evstartsetup**

**tminserttime**

Flows:

**informations\_check**

From informations to check

**check\_Annexlab**

From check to Annexlab

Globals:

Classes:

**Annex\_Management\_system**

Objects:

**student\_detail**

Implicit, Multiplicity of 1

**systems\_detail**

Implicit, Multiplicity of 1

Operations:

**check**

Primitive-operation , Public, Return type is void

**evclose**

Reception

**evopen**

Reception

**evstart**

Primitive-operation , Public, Return type is void

**evstartsetup**

Reception

**insert**

Primitive-operation , Public, Return type is void

**remove**

Primitive-operation , Public, Return type is void

**setup**

Primitive-operation , Public, Return type is void

**tminserttime**

Reception

Attributes:

**atime**

Type of float, Public, Multiplicity of 1

**btime**

Type of float, Public, Multiplicity of 1

**nomber\_systems**

Type of int, Public, Multiplicity of 1

**rollnomber**

Type of int, Public, Multiplicity of 1

**sid**

Type of int, Public, Multiplicity of 1

Statechart

****

**Overridden Properties**

Subjects:

**Format**

Metaclasses:

**DiagramConnector**

Properties:

Line.LineWidth: 1

Font.FontColor: 0,0,0

Fill.FillColor: 152,158,231

Font.Font: Arial

DefaultSize: 0,0,55,24

Font.Size: 10

Line.LineColor: 0,0,0

**HistoryConnector**

Properties:

Line.LineWidth: 1

Font.FontColor: 0,0,0

Fill.FillColor: 152,158,231

Font.Font: Arial

DefaultSize: 0,0,25,23

Font.Size: 10

Line.LineColor: 0,0,0

**DanglingTransition**

Properties:

Line.LineWidth: 2

Line.LineColor: 0,0,0

**State**

Properties:

Line.LineWidth: 1

Font.FontColor: 0,0,0

Fill.FillColor: 255,255,255

Font.Font: Arial

DefaultSize: 0,0,96,60

Line.LineStyle: 0

Font.Size: 10

Line.LineColor: 0,112,0

**ROOT**

Or-state

Substates:

close

open

Running

Default Transition

Target:

close

**close**

Or-state

Out Transition

evstartsetup

Target:

Running

**open**

Or-state

Out Transition

evclose

Target:

History Connector

**Running**

Or-state

Substates:

checkdetail

inserting

studentdetail

Out Transition

evopen

Target:

open

Default Transition

Target:

studentdetail

**checkdetail**

Or-state

EntryAction

System.out.println("checking");

ExitAction

System.out.println("details");

Out Transition

tm(checktime)

Target:

inserting

**inserting**

Or-state

EntryAction

System.out.println("Door Opened");

ExitAction

System.out.println("Door Closed");

Out Transition

tm(inserttime)

Target:

done

**studentdetail**

Or-state

EntryAction

System.out.println("studentdetail");

Out Transition

tm(detailttime)

Target:

checkdetail

**done**

Diagram Connector

Out Transition

Target:

close

**done**

Diagram Connector

**display**

Operations:

**disployinformation**

Primitive-operation , Public, Return type is void

**empty\_full**

Primitive-operation , Public, Return type is void

**evKeyprocess**

Reception

**evpress**

Primitive-operation , Public, Return type is void

**process**

Primitive-operation , Public, Return type is void

Statechart

****

**Overridden Properties**

Subjects:

**Format**

Metaclasses:

**StateChartConditionConnector**

Properties:

Line.LineWidth: 1

Font.FontColor: 0,0,0

Fill.FillColor: 152,158,231

Font.Font: Arial

DefaultSize: 0,0,25,23

Font.Size: 10

Line.LineColor: 0,0,0

**DanglingTransition**

Properties:

Line.LineWidth: 2

Line.LineColor: 0,0,0

**State**

Properties:

Line.LineWidth: 1

Font.FontColor: 0,0,0

Fill.FillColor: 255,255,255

Font.Font: Arial

DefaultSize: 0,0,96,60

Line.LineStyle: 0

Font.Size: 10

Line.LineColor: 0,112,0

**ROOT**

Or-state

Substates:

Waitforkeys

Default Transition

Target:

Waitforkeys

**Waitforkeys**

Or-state

Out Transition

evKeyprocess

Condition Connector

Branches:

[params.key == 's']/itsDishwasher.gen(new evStart());

Target:

Waitforkeys

[params.key == 'o']/itsDishwasher.gen(new evOpen());

Target:

Waitforkeys

[params.key == 'c']/itsDishwasher.gen(new evClose());

Target:

Waitforkeys

[params.key == 'x']/System.exit(1);

Target:

Waitforkeys

**Keyreader**

**Overridden Properties**

Subjects:

**CG**

Metaclasses:

**Class**

Properties:

Concurrency: active

Actors:

**serviceman**

Association Ends:

**itsService**

Association with service, Multiplicity of 1, Bi-directional, Navigable

**staf**

Association Ends:

**itsCheck**

Association with check, Multiplicity of 1, Bi-directional, Navigable

**student1**

Association Ends:

**itsInformations**

Association with informations, Multiplicity of 1, Bi-directional, Navigable

**student2**

Association Ends:

**itsInformations**

Association with informations, Multiplicity of 1, Bi-directional, Navigable

Use Cases:

**Annexlab**

SubUseCases:

check

**check**

Association Ends:

**itsStaf**

SuperUseCases:

Annexlab

Public

SubUseCases:

informations

**informations**

Association Ends:

**itsStudent1**

**itsStudent2**

SuperUseCases:

check

Public

**service**

Association Ends:

**itsServiceman**

Default

Language: C++

Globals:

Object Model Diagrams

**Model1**

****Components

DefaultComponent

Language: C++

Component Settings:

Build type: Executable

Configurations:

**DefaultConfig**

Scope type: Explicit

Instrumentation type: None

Time-model type: Real-time

Statechart generation type: Flat

Files and Folders: