Basic design – Design Document

for

Home Project

Version 1.1

Written by Pham Nhat Minh Tan
Instructed by Pham Ba Thanh (Mentor)

HISTORY

Version	Author	Inspector	Date	Description
1.1	TanPNM	ThanhPB	17/04/2021	1. Add Class Diagram 2. Update Flow design
1.0	TanPNM	ThanhPB	19/03/2021	Create Document

Table of Contents

I. User Interface Design

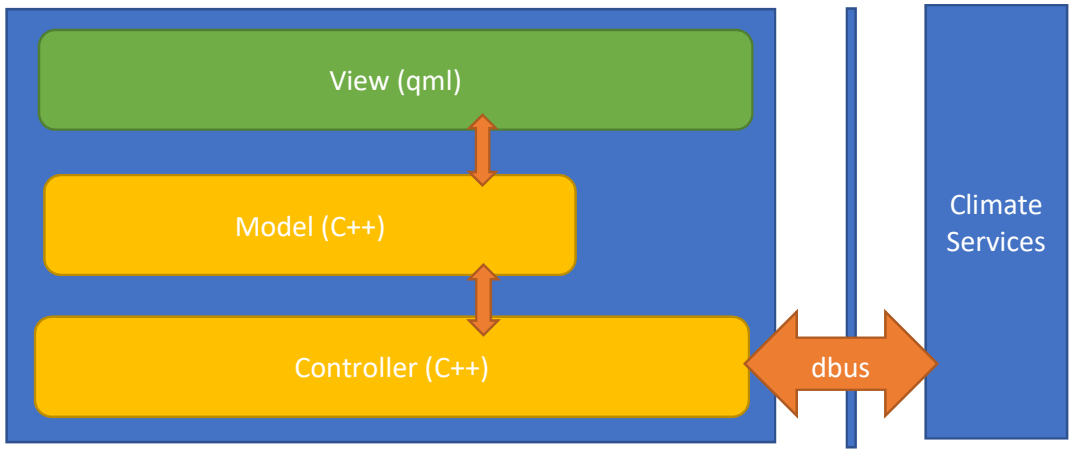
Refer UI document: AAD305_UI_PhamNhatMinhTan_FX000659.pdf

II. User Experience Design

Refer UX document: AAD305_UX_PhamNhatMinhTan_FX000659.pdf

III. Architecture Design

1. Overall diagram of application

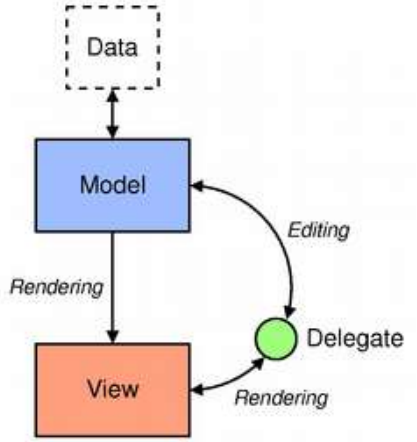


View (qml): This is where the screens are managed, components built with qml and the resources of the screen building.

Model: As the data building for managing the state of the interface from C ++, it is the place to show the data for the state construction of the screen.

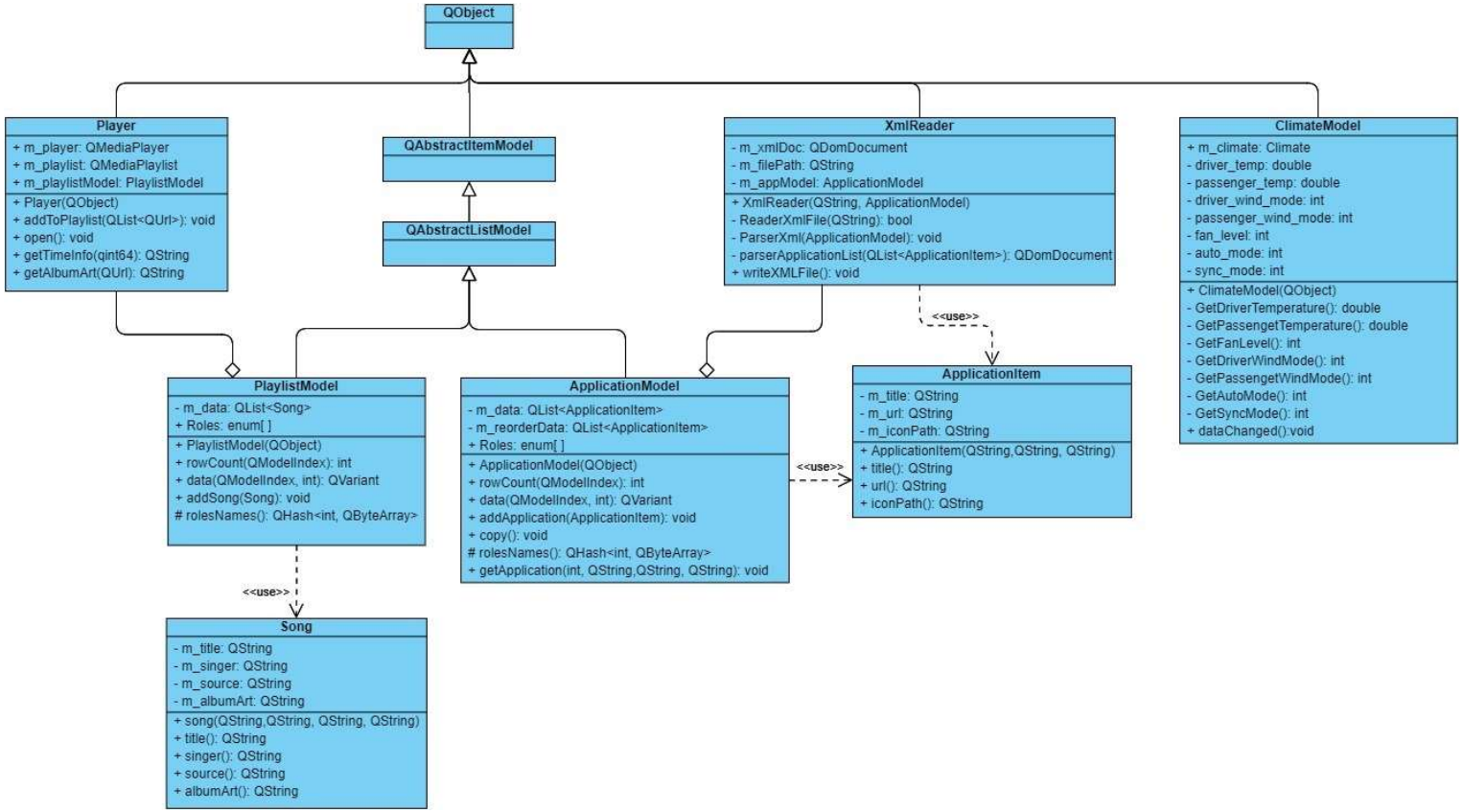
Controller: As part of processing, program control, and responsible for connecting to the 3rd services (specifically here is climate services).

The architecture built for the program is built on the Model View architecture



Data: The xml contains information of all applications in the system.
Model: Class stores application list read from xml file.
View: QML displays a list of applications.

IV. Class Diagram



Description classes:

1. Player

Function	Description	Input	Output
Player	Constructor	QObject	
addToPlaylist	Add song to Playlist	QList<QUrl>	
open	Open music folder to add song		
getTimeInfo	Format time input to format mm:ss	qint64	QString
getAlbumArt	Get album art of the song	QUrl	QString

2. PlaylistModel

Function	Description	Input	Output
PlaylistModel	Constructor	QObject	
rowCount	Count item in model	QModelIndex	int
data	Get roles of model	QModelIndex Int	QVariant
addSong	Add song to model	Song	
roleNames	Set role to model		QHash<int, QByte>

3. Song

Function	Description	Input	Output
Song	Constructor	QString QString QString QString	
title	Get title		QString
singer	Get singer		QString
source	Get source		QString
Album_art	Get album art		QString

4. ApplicationItem

Function	Description	Input	Output
ApplicationItem	Constructor	QString QString QString	
title	Get title		QString
url	Get url		QString
iconPath	Get path of icon		QString

5. ApplicationModel

Function	Description	Input	Output
PlaylistModel	Constructor	QObject	
rowCount	Count item in model	QModelIndex	int
data	Get roles of model	QModelIndex Int	QVariant
addApplication	Add application to model	ApplicationItem	
getAppsList	Get application list after reorder an application		QList<ApplicationItem>
copy	Copy default app list to app list after reorder		
roleNames	Set role to model		QHash<int, QByte>
getApplication	Get application from qml to add to QList	Int QString QString QString	

6. XmlReader

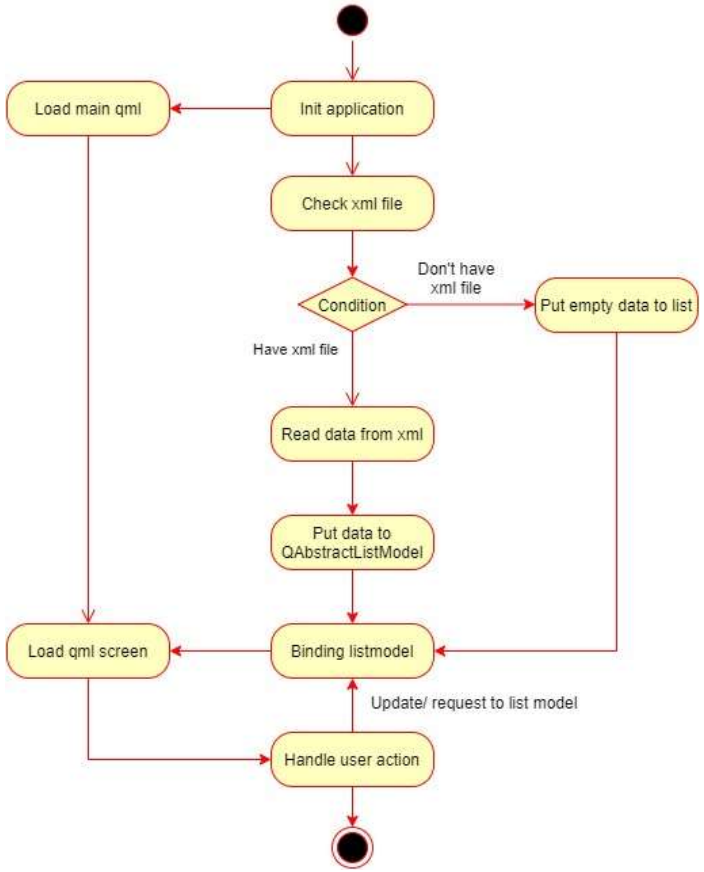
Function	Description	Input	Output
XmlReader	Constructor	QString ApplicationModel	
ReadXmlFile	Read XML File	QString	bool
PaserXml	Convert data in XML file to application model	ApplicationModel	
parserApplicationList	Convert application model to XML	QList<ApplicationItem>	QDomDocument
writeXMLFile	Save XML data to file		

7. ClimateModel

Function	Description	Input	Output
ClimateModel	Constructor	QObject	
GetDriverTemperature	Get driver temperature		double
GetPassengerTemperature	Get passenger temperature		double
GetFanLevel	Get fan level		int
GetDriverWindMode	Get driver wind mode		int
GetPassengerWindMode	Get passenger wind mode		int
GetAutoMode	Get auto mode		int
GetSyncMode	Get Sync mode		int

V. Processing Flow Design

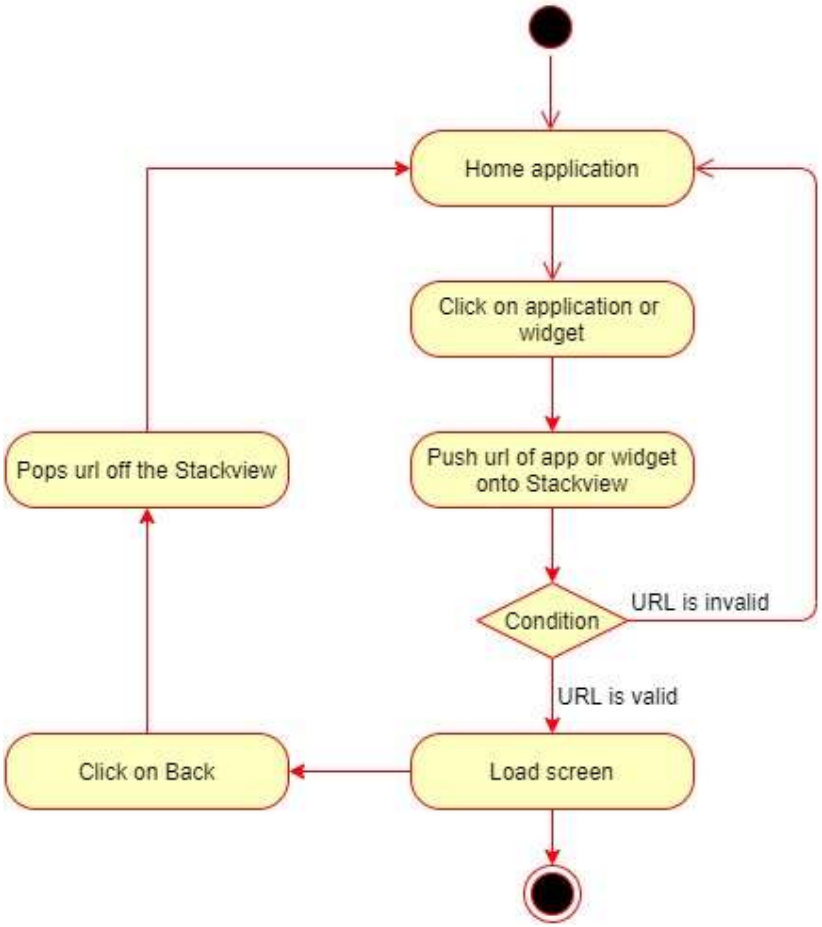
1. Processing flow when starting the Home application



Steps to start home program:

Step 1: Create engine object of QqmlApplicationEngine
Step 2: Create appsModel object of ApplicationsModel
Step 3.4: Create xmlReader object of XmlReader with the value passed as the path to the xml file and appsModel object
Step 5: Read the xml file
Step 6: Parse information from xml to ApplicationsModel object
Step 7: Binding appsModel to QML by settingContextProperty
Step 8: Start the QML engine by loading the url of the main qml file

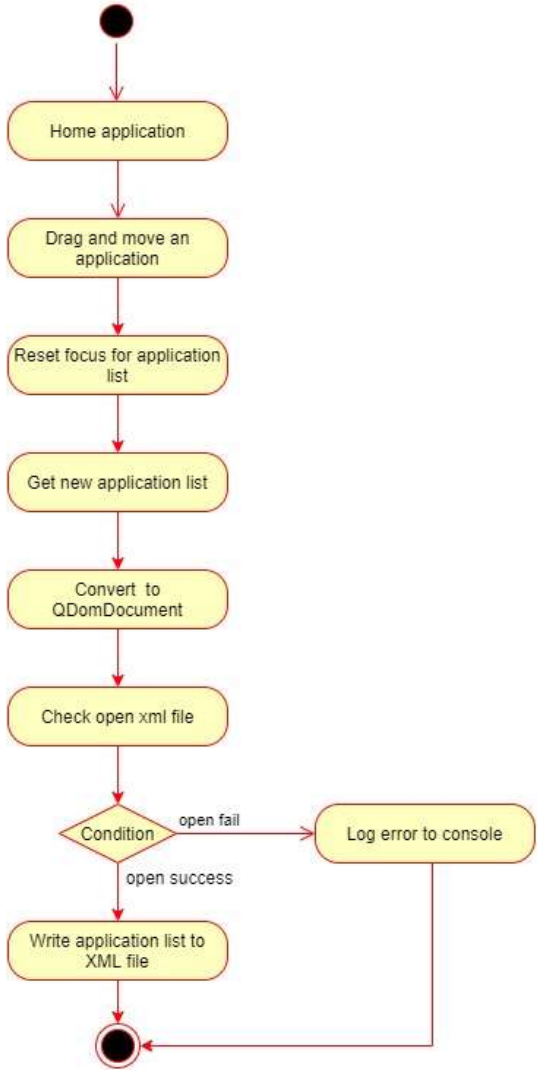
2. Processing flow when open and close application or widget



Steps to open and close application or widget:

Step 1: Open Home application
Step 2: Click on any application or widget
Step 3: Push QML file path of application or widget onto Stackview
Step 4: Load QML screen
Step 5: Click on Back button
Step 6: Pops QML file off Stackview
Step 7: Load Home Screen

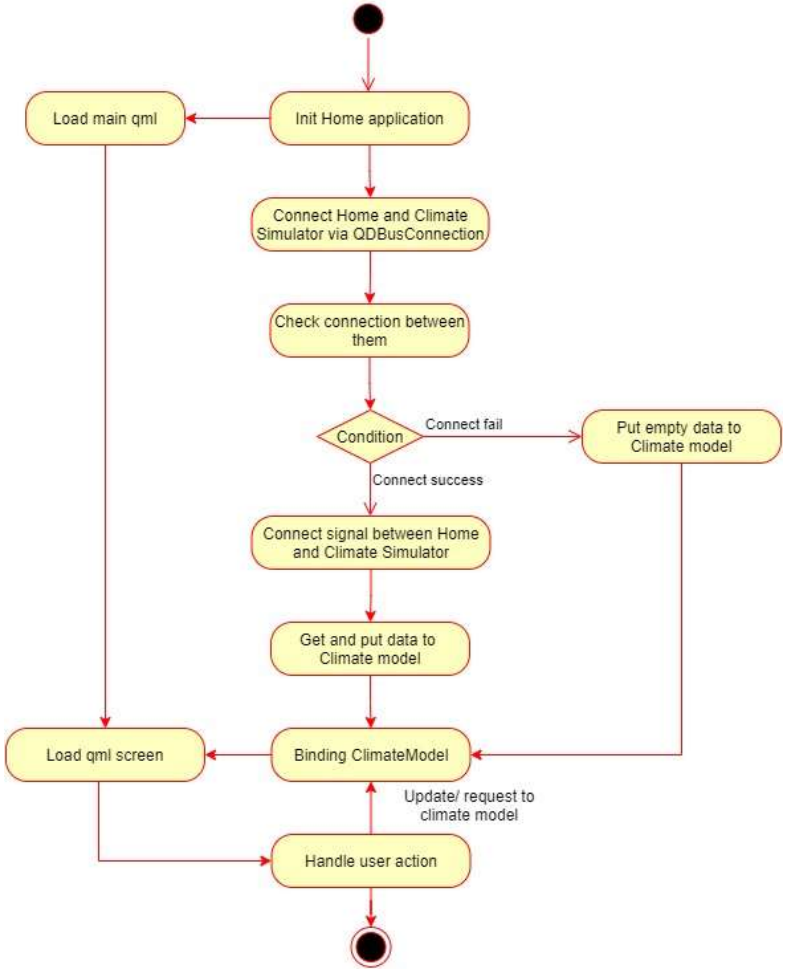
3. Processing flow when reorder application list



Steps to reorder application list:

Step 1: Open Home application
Step 2: Drag and move an application
Step 3: Traversal the list to reset focus state for new application list
Step 4: Get each application in model to store into new list
Step 5: Convert new list model to QDomDocument with format XML file
Step 6: Check open “application.xml” file successful or not
Step 7: Write the QDomDocument is converted to application.xml file

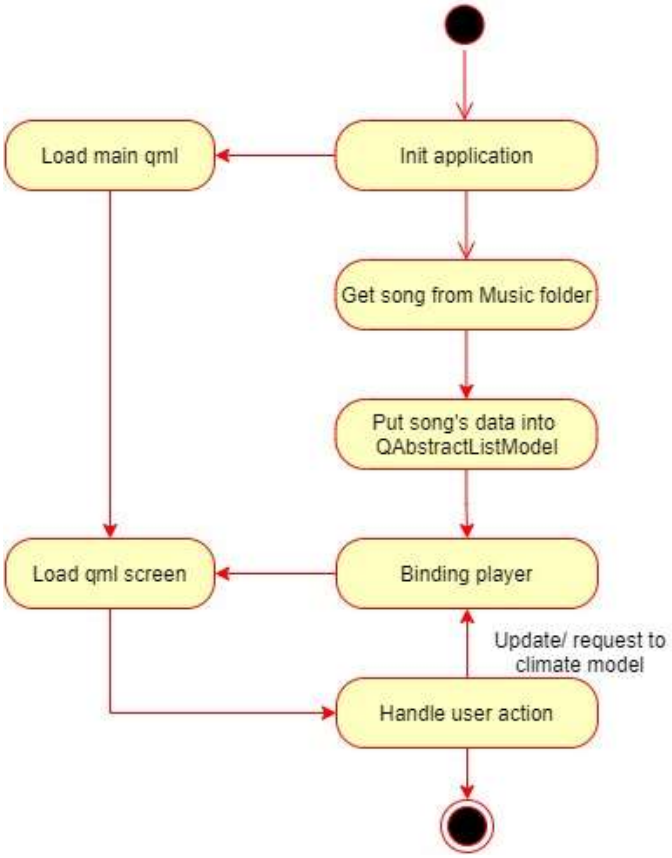
4. Processing flow of send information from Climate simulator to Climate widget



Step to send information from Climate simulator to Climate widget:

- Step 1: Create engine object of QqmlApplicationEngine
- Step 2: Create climate object of ClimateModel
- Step 3: Connect ClimateModel object on Home with Climate object on Climate simulator via QDBusConnection
- Step 4: Connect signal between Climate simulator and Home application
- Step 5: Get data from Climate simulator and put it to ClimateModel object
- Step 6: Binding ClimateModel to QML by settingContextProperty
- Step 7: Start the QML engine by loading the url of the main qml file

5. Processing flow of send information from Media application to Media widget



Step to send information from Media application to Media widget:

Step 1: Create engine object of QqmlApplicationEngine

Step 2: Create player object of Player

Step 3: Initialize QMediaPlayer, QMediaPlaylist and PlaylistModel object in Player

Step 4, 5: Get song from Music folder and put data of the songs to PlaylistModel object

Step 6: Binding player to QML by settingContextProperty

Step 7: Start the QML engine by loading the url of the main qml file