Software Design Specification

For

Home Application

Prepared by : Ho Khanh Vu – SE 05427

Organization: FPT Education

Date created: 15/08/2021

Revision History

Date	Description	Verson
15/08/2021		1.0

Table of Contents

I. Introduction

1. Purpose

This document will outline in detail the software architecture and design for the Home Application. This document will provide several views of the tool's design in order to facilitate communication and understanding of the app. It intends to capture and convey the significant architectural and design decisions that have been made for the Home Application.

2. Definitions, Acronyms and Abbreviations

SDS: Software Design Specification.
UI Design: User Interface Design
UX Design: User Experience Design

3. Intended Audience and Reading Suggestions

FUNiX, Developer, Tester

4. Project Scope

This document defines architecture and design for developing Home application.
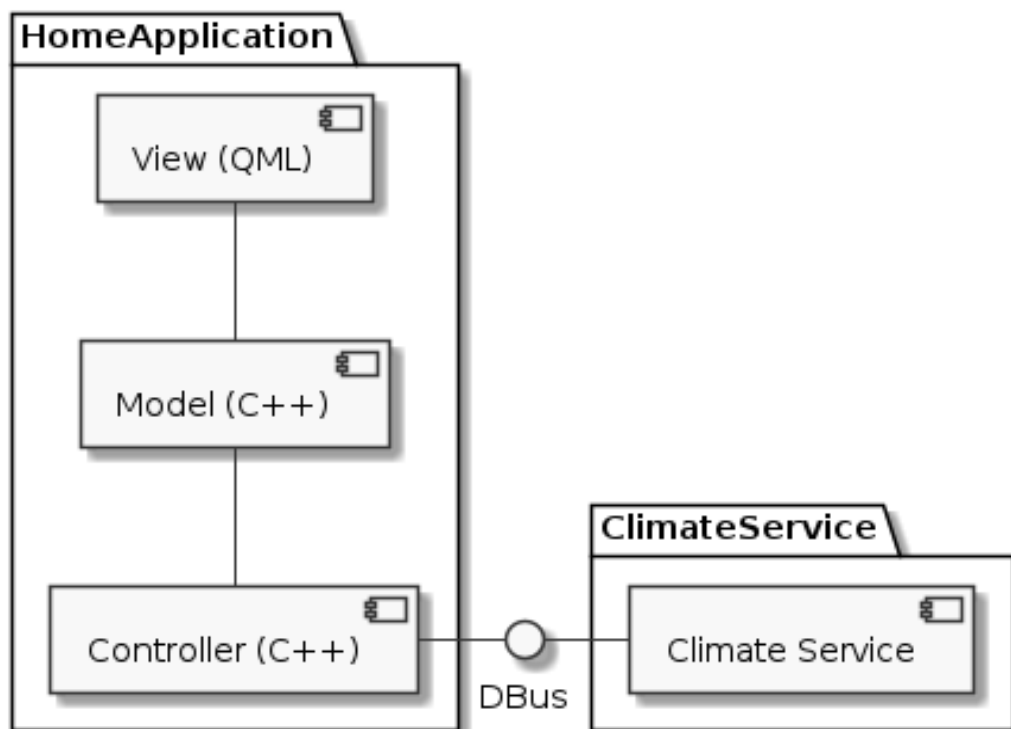
II. UI Design

Attachment

III. UX Design

Attachment

IV. Architecture Design

Component	Descriptions
View (QML)	The place where application controls screens, components that are built from QML and resources of building screens.
Model	The place where is built interface for state management of interface from C++. It is where the data for building state of screen is shown.
Controller	Controls, executes the program and takes responsibility for connecting the third-party services (in this case, it is climate services).

The architecture built for the application is based on Model View architecture.
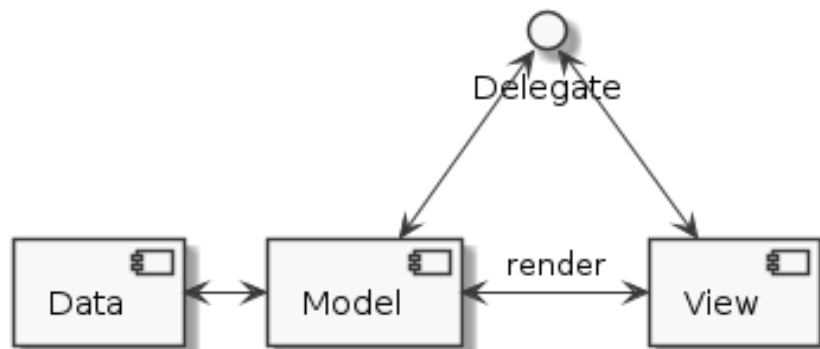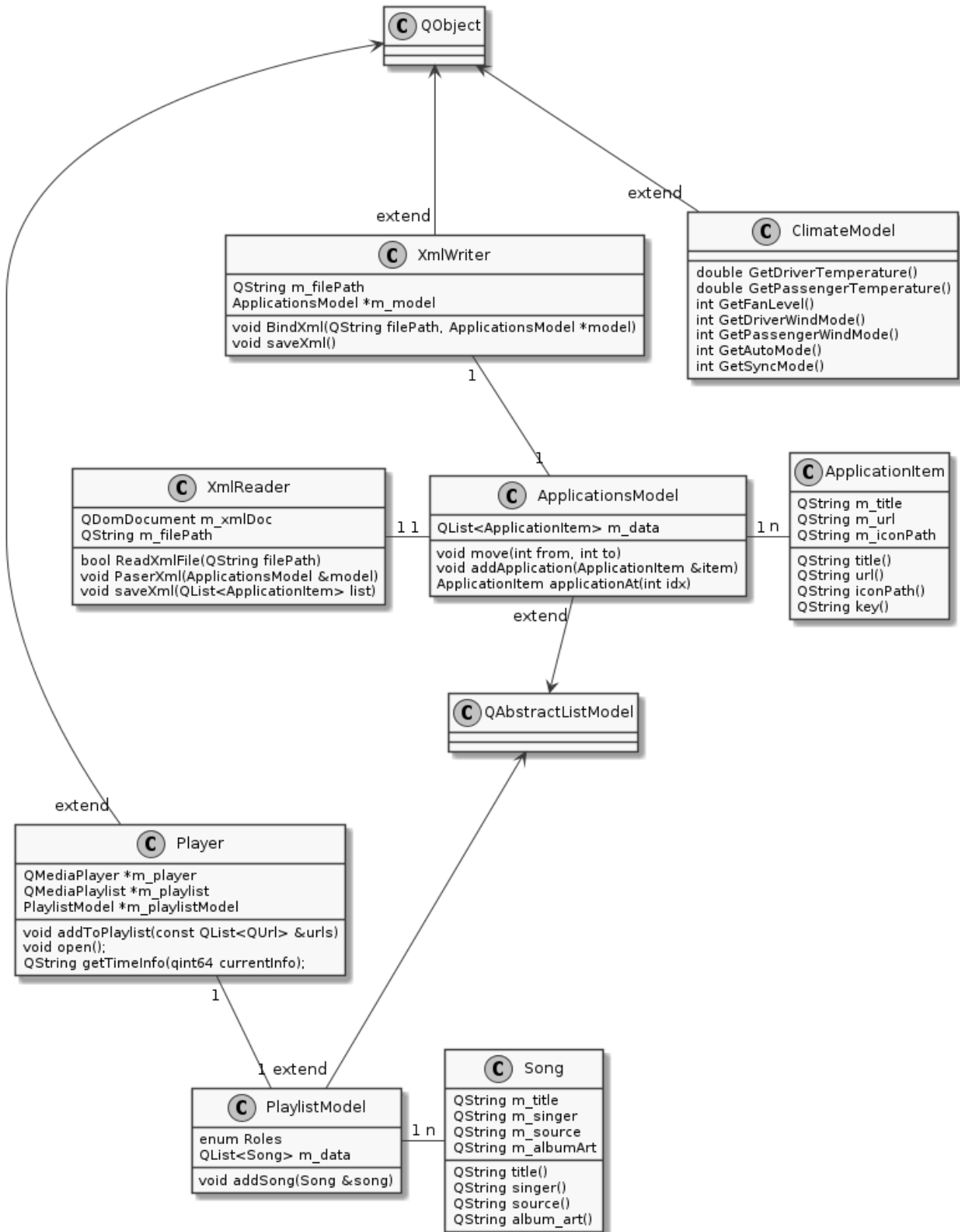

Data: Xml stores application information in system.

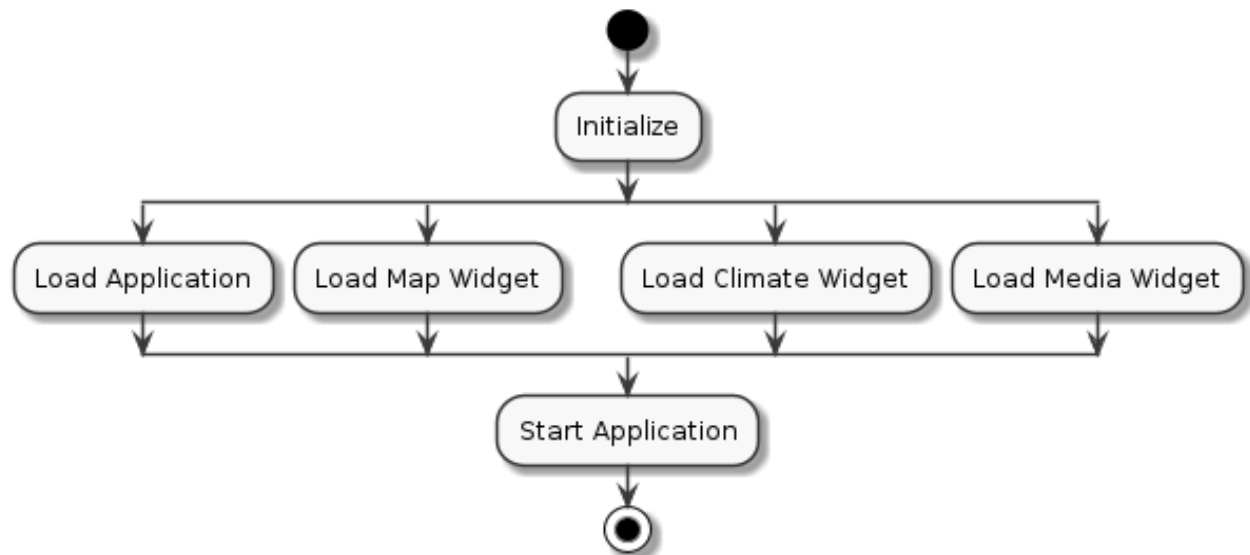Model: Class stores application list that is read from ".xml" file

View: QML displays application list.

V. Class Diagram

QObject

XmlWriter
extend

QString m_filePath
ApplicationsModel *m_model

void BindXml(QString filePath, ApplicationsModel *model)
void saveXml()

ClimateModel
extend

double GetDriverTemperature()
double GetPassengerTemperature()
int GetFanLevel()
int GetDriverWindMode()
int GetPassengerWindMode()
int GetAutoMode()
int GetSyncMode()

XmlReader

QDomDocument m_xmlDoc
QString m_filePath

bool ReadXmlFile(QString filePath)
void PaserXml(ApplicationsModel &model)
void saveXml(QList<ApplicationItem> list)

ApplicationsModel

QList<ApplicationItem> m_data

void move(int from, int to)
void addApplication(ApplicationItem &item)
ApplicationItem applicationAt(int idx)

ApplicationItem

QString m_title
QString m_url
QString m_iconPath

QString title()
QString url()
QString iconPath()
QString key()

QAbstractListModel
extend

Player
extend

QMediaPlayer *m_player
QMediaPlaylist *m_playlist
PlaylistModel *m_playlistModel

void addToPlaylist(const QList<QUrl> &urls)
void open();
QString getTimeInfo(qint64 currentInfo);

PlaylistModel
extend

enum Roles
QList<Song> m_data

void addSong(Song &song)

Song

QString m_title
QString m_singer
QString m_source
QString m_albumArt

QString title()
QString singer()
QString source()
QString album_art()

VI. Flow Chart

1. Starting Home Application

Order	Description
1	Initialize Home application
2.1	Load Applications from XML
2.2	Load Map Widget
2.3	Load Climate Widget
2.4	Load Media Widget
3	Start Application

2. Load List Application
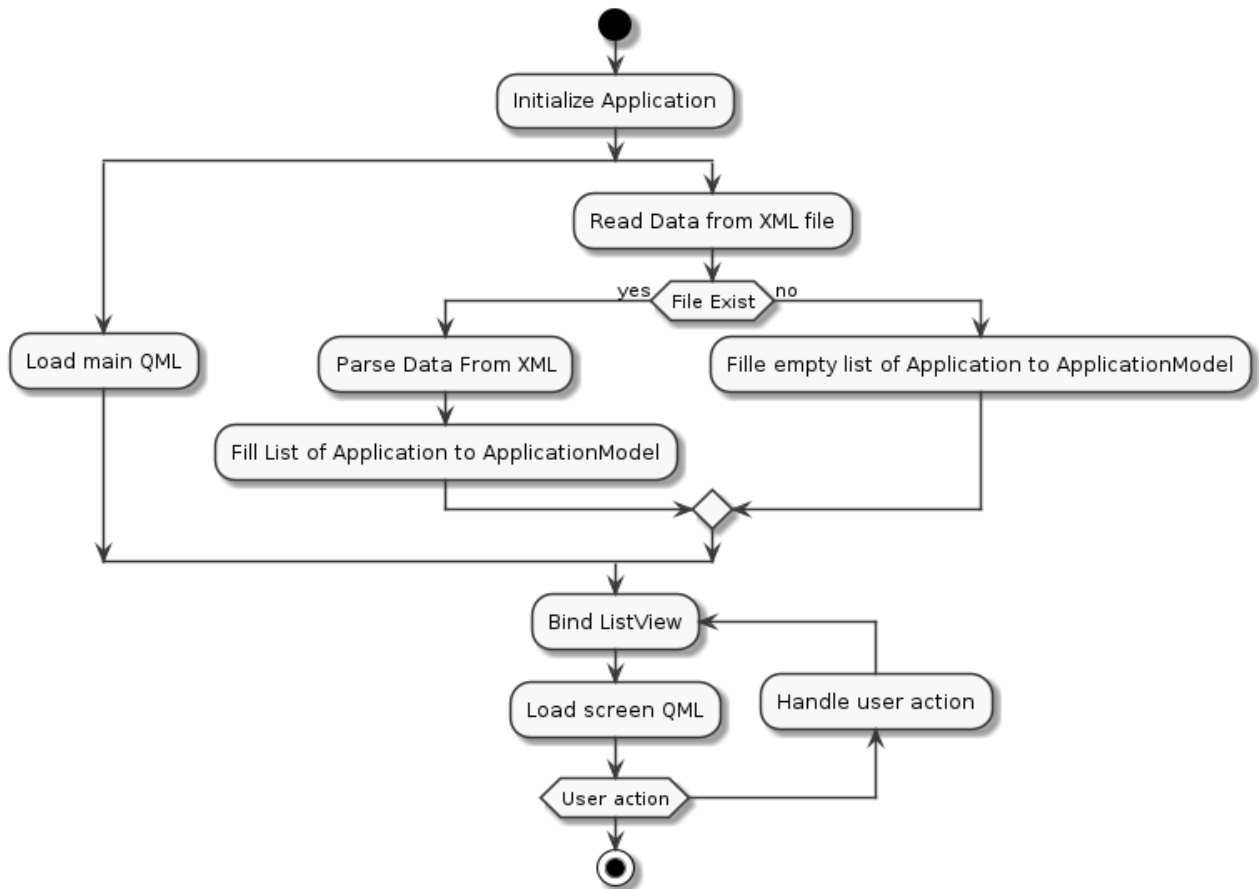
Order	Description
1	Initialize Home Application
2.1	Load Main QML
2.2	Read applications from XML file
3.1	If XML file does not exist, go to 5
3.2	If XML file does exist, parse data
4	Fill list parsed applications to application model
5	Bind application model to list view
6	Load Screen QML
7	Waiting for User Action
8	Handle user action, go to 5

3. Open Application / Widget
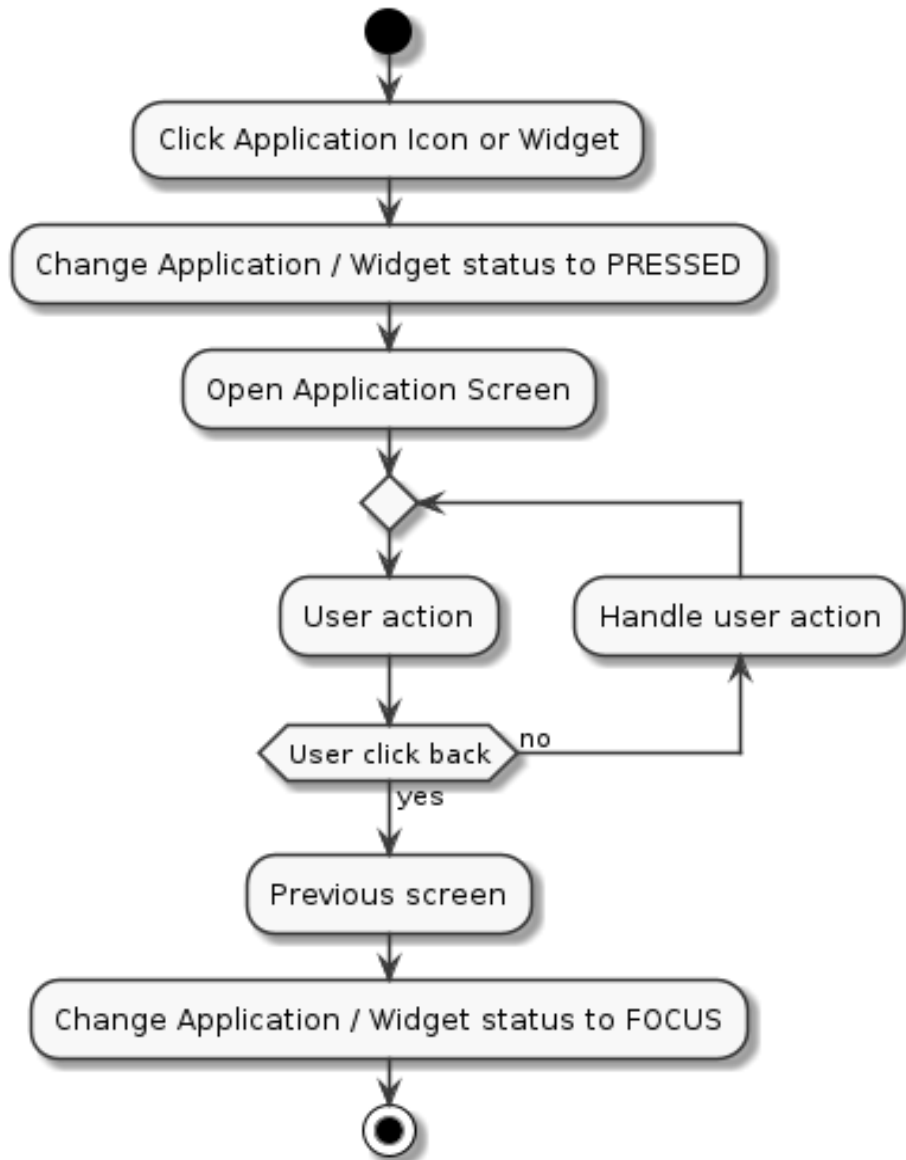
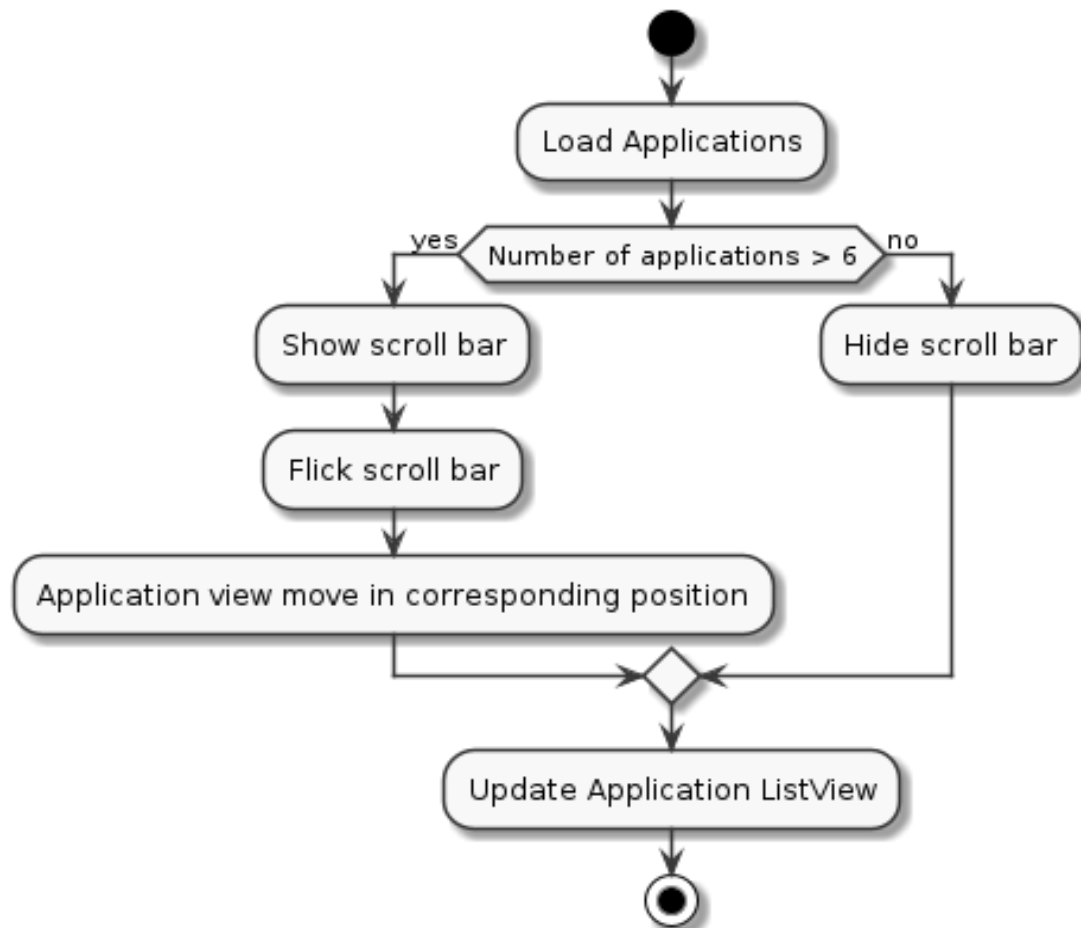| Order	Description
1	Click on application / widget icon
2	Change application / widget status to PRESSED
3	Open application screen
4	Wait for user action
5.1	If user does not click back button, handle action and go to 4
5.2	If user click back button, go to 6
6	Go to previous Screen
7	Change application / widget status to FOCUS

4. Communicate with Climate Simulator
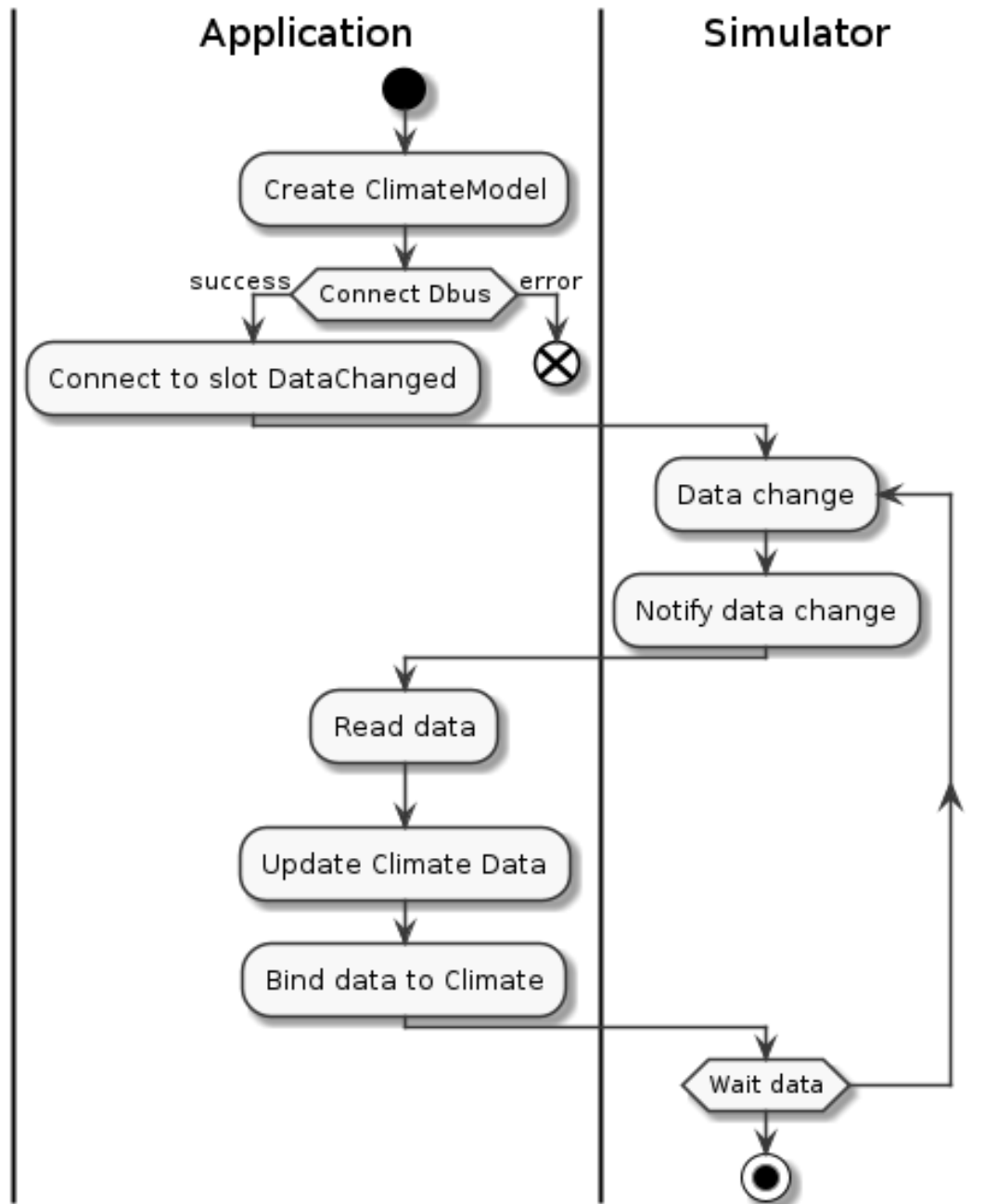
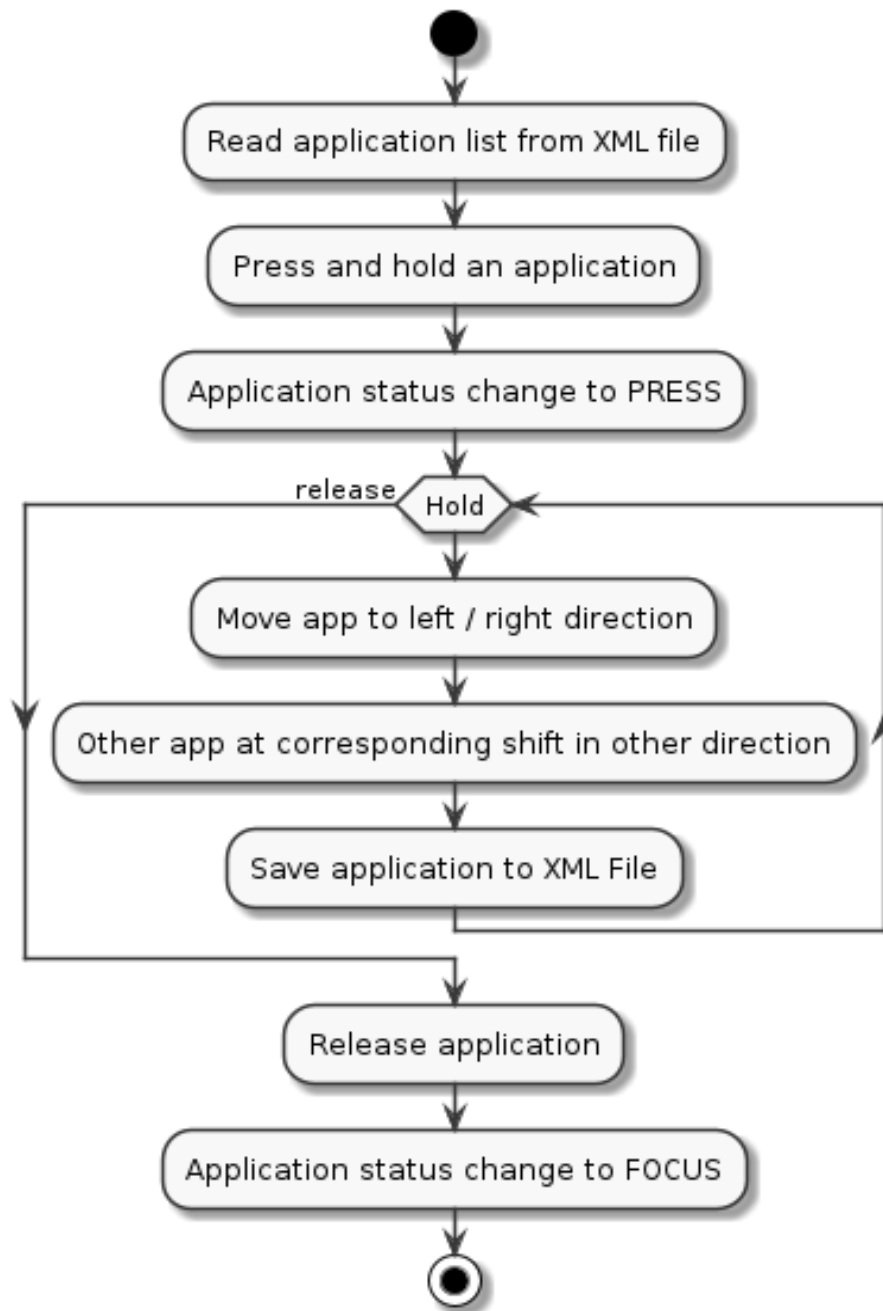| Order	Description
1	Create climate model
2.1	If connect to Dbus error, end
2.2	If connect to Dbus success, go to 3
3	Connect to slot DataChanged
4	Climate data change
5	Climate notify data changed using Dbus
6	Home application read data

7	Update climate data
8	Bind data to climate
9	Waiting to data, if data changed go to 4

5. Scroll Application List

Order	Description
1	Load application
2.1	If Number of application less or equal 6, go to 3
2.2	If Number of application more than 6, go to 4
3	Hide scroll bar, go to 6
4	Show scroll bar
5	Application view move into corresponding position
6	Update application ListView

6. Rearrange Application order

Order	Description
1	Read application list from XML
2	Press and hold on application icon
3	Application status change to PRESS
4.1	If keep holding, go to 5
4.2	If release, go to 8

5	Move app to left / right direction
6	Other app at corresponding move to opposite direction
7	Save application to XML, go to 4
8	Release application
9	Application status change to FOCUS