A Location-Based Movie Advisor Application for Android Devices

Taner Arsan, Aykut Çayır, Hande Nur Umur, Tuğçe Güney, and Büke Panya

Abstract

Android is one of the world's most popular mobile platforms. There are more than 600,000 applications available today's market place. Movie advisor applications are also available in Google Play, but there is no location-based movie advisor application for Android devices in Google Play and any other marketplace. A Location-Based Service is a mobile computing application that provides information and functionality to users based on their geographical location. In this study, a location-based movie advisor which is a special application for Android devices to find nearest movie theaters, is developed and implemented. Android devices are getting smarter with new features. By using these devices, we can use new technologies and new ideas. Location-based services are one of these ideas. Wherever you are, you can search and find new possibilities for almost everything. The aim of the location-based movie advisor application for Android devices is to give a brief summary about movies, movie times and also nearest location information of the movie theaters depending on the location of the user.

Keywords

Software architecture • Mobile application development • Android programming • Location-based service

Introduction

In mobile computing applications, a location based service provides services to users based on their location. Geographical location information can be used to find the closest market places or pharmacies and so on. First generation location based services (LBS) were designed as a client-server architecture. For instance, users asked an application for location information and then received a response. In 1996, The White House authorized that selective availability of Global Positioning System (GPS) signals would be phased out [1]. Location based services were standardized by Java Micro Edition Release for mobile platforms which supported Java Platform. With announcement of iPhone, the big revolution in mobile computing came out in June 2007. Almost 1 year later, an Android phone which is known as the greatest rival of iPhone was released by HTC, in October 2008 [2]. Some examples of location based services can be listed as:

- Recommending social events in a city [3].
- Requesting the nearest business or service, such as an ATM or restaurant.
- Turn by turn navigation to any address.
- Locating people on a map displayed on the mobile phone.
- Receiving alerts, such as notification of a sale on gas or warning of a traffic jam.
- Location-based mobile advertising.
- Asset recovery combined with active RF to find, for example, stolen assets in containers where GPS would not work.

T. Arsan (✉) • A. Çayır • H.N. Umur • T. Güney • B. Panya
Department of Computer Engineering, Kadir Has University, Cibali, 34083 Istanbul, Turkey
e-mail: arsan@khas.edu.tr

T. Sobh and K. Elleithy (eds.), *Innovations and Advances in Computing, Informatics, Systems Sciences, Networking and Engineering*, Lecture Notes in Electrical Engineering 313,
DOI 10.1007/978-3-319-06773-5_27, © Springer International Publishing Switzerland 2015

- Games where your location is part of the game play, for example your movements during your day make your Avatar move in the game or your position unlocks content.
- Real-time Q&A revolving around restaurants, services, and other venues.

In Android platform, there are two kinds of location-based services. One of them is “Android’s Network Location Provider” and the other is “GPS utilization”. GPS is more accurate than network location provider. However it is convenient for outdoor usage only. It also consumes battery power and has got latencies. Android’s Network Provider uses Wi-Fi signals. It works indoors or outdoors, uses less battery power and return the location information as quickly as users want [4]. In practice, a programmer can use both GPS and Android’s Network Providers.

There are several problems for obtaining user’s location from a mobile device [5]. Reasons are

- multitude of location sources,
- user movement,
- varying accuracy

Next generation location-based services provide additional benefits for users and service providers. There is an explosive growth in the LBS market over the coming years. Apart from the consumer market, there is a good potential for this technology to find its application in many industries including health, manufacturing, mining and financial services [1].

This study is related with a location-based movie advisor application for Android devices which aims to give brief summaries about movies, movie times and also nearest location information of the movie theaters depending on the location of the user. The rest of this paper is organized as follows. In section “Project Specifications”, we explain the project specifications and then in section “Determination of Requirements” we determine project requirements. So, we give the all details of software architecture design stages and propose a new location-based mobile application in section “Three-Tier Software Architecture for Proposed Location Based Movie Advisor”. In section “Implementation”, we show the key idea and implementation steps. In section “Complete System Architecture”, we present the complete system architecture, and conclusions are then given in section “Conclusions”.

Project Specifications

In this paper, the aim is to develop a location-based Android application that give a brief summary about movies, timetables of the movies and also nearest location information of the movie theaters depending on the location of the user. Briefly, the location-based movie advisor shows the movie theaters according to distance in mobile phones and tablets. This application is called as Cinephilia (love of cinema). Cinephilia has the following specifications: “Cinephilia considers only information for movie theaters of shopping centers in Istanbul-Turkey. Cinephilia considers only today’s movies. Cinephilia provides movie times, brief summaries, rating scales of movies, and location of movie theaters.”

In today’s technologies there are some movie advisor software and web sites are available, but Cinephilia has significant differences. When users start using Cinephilia, their location is determined according to their area. According to this location information, Cinephilia shows nearest movie theaters of shopping centers in Istanbul. In addition, users can learn today’s movies and upcoming movies, and users can search movies and get brief information about movies, location of the shopping centers, contact information of movie theaters and finally rating scales of movies and links to movie trailers. Moreover, users give rates to movies, and users get information about the cast. Finally, we can say that Cinephilia has a user friendly graphic user interface and screen transitions.

Determination of Requirements

System Requirements

Cinephilia is an application for Android devices, so it is necessary to use a smart phone or a tablet with Android 2.2 operating system or higher version. Android device should be connected to Internet using IEEE 802.11b/g/n or 3G MSDPA. Android device should have minimum 2Mbyte of free memory and 320 × 480 pixels screen resolution with 3.5 in. screen.

User Requirements

Wi-Fi or 3G Internet connection should be started up by the users. The application should be started by the users. The users can see all the names of movie theaters and its movie times in Istanbul shopping centers as a list. The users can follow future programs. The users can see name of movie theaters when they permit to determine their location.

Three-Tier Software Architecture for Proposed Location Based Movie Advisor

Cinephilia is the three-tier software architecture as shown in Fig. 1. It is a total solution for accessing all information about movies and it is for people who are interested in

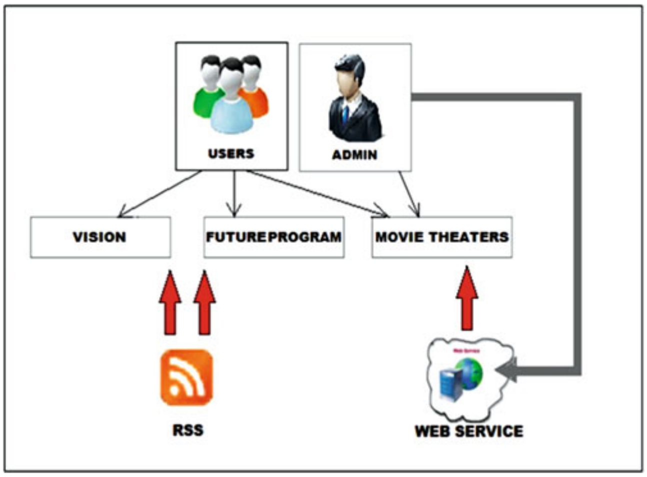


Fig. 1 Three-tier software architecture for location-based movie advisor

movies. Cinephilia includes different sections which are VISION, FUTURE PROGRAM, and MOVIE THEATERS for all users. These sections are independent from each other. There are two different type of User in Cinephilia application.

These are System Administrator and Application Users. The software includes JSON web service and a web feed which is a RSS feed. Web service and RSS feeds provide to access easily actual movies and movie theaters data to users.

System Administrator writes the web service. Administrator can add new movie theaters to web service. System Admin assigns an ID of each movie and adds names of the movies and location of movie theaters according to latitude and longitude variables into web service. Page contents are created according to all cinema ID's in web service. Every content page include only one cinema ID and the page include information about movies and session hour according to this cinema ID. Admin has authority that adding new movies or deleting movies and information about movies. User can see all steps of application. However they cannot do any changes to the movie theaters. Use case diagram of the Location-based Movie Advisor is shown in Fig. 2.

There are several steps for program execution. When Cinephilia splash screen is opened, users see elephant photo with toast message which is "Loading". In totally, splash screen is appeared 18 s and "Loading" toast message lasts 5 s. While these were on the screen, a melody is playing at the background. After finishing the splash screen, the vision page is now open with three tabs which are on the top. While these applications are working, two RSS are parsed in behind. Also, application includes simple web service with xml and http. It is easy to convert web services from web applications. Then, two web services are found for application. For the first address for visions; http://rss.beyazperde.com/movieler/bunlar?format=xml and for future program is founded http://rss.beyazperde.com/movie-ler/buhafta?format=xml address. We have to parse the xml. There are three tabs for users that are vision, this week and cinema halls. All transactions are described in the following section.

Vision Screen

Vision screen which is given in Fig. 3 is the first tab for our application and shows users to see past and present movies in vision. In this screen, the user has a chance to see more detailed information about movies such as movie subject, director name, player name, duration time and its type. Output screen is as follows:

1. If vision screen is requested to open, user should touch vision page from first page screen.
2. When users want to achieve more information about movies which are movie subject, date display and players, they should select any movie and click on it. Then, detail page is opened.

This Week Screen

The user can follow the information about current movies information on this week screen page. By using this screen, wide information obtained from a page which is opened after clicking any movie which you select. Explanations for the subject of the movie, director's name, cast (players), duration time and its type are seen in this page. Output screen is as follows:

1. If this week screen is wanted to open, user should click this week page from first page screen.
2. When users interested in any movie and want to see more information about movies which are subject of the movie, date displayed and players, they should select any movie and click on it. Then, detail page is opened.

Movie Theaters Screen

When the user clicks "Movie Theaters", several steps are achieved in the background. Firstly, latitude and longitude of locations, positions of the users, are listed in the screen. Names of the movie theaters are shown with their IDs and longitude and latitude values which are we entered from http://cineapi.herokuapp.com/theatres address for users can see them. Then, http://cineapi.herokuapp.com/theatre/1/can check the information about movie times and movies which parsed from the web page of sinemalar.com. After that, users

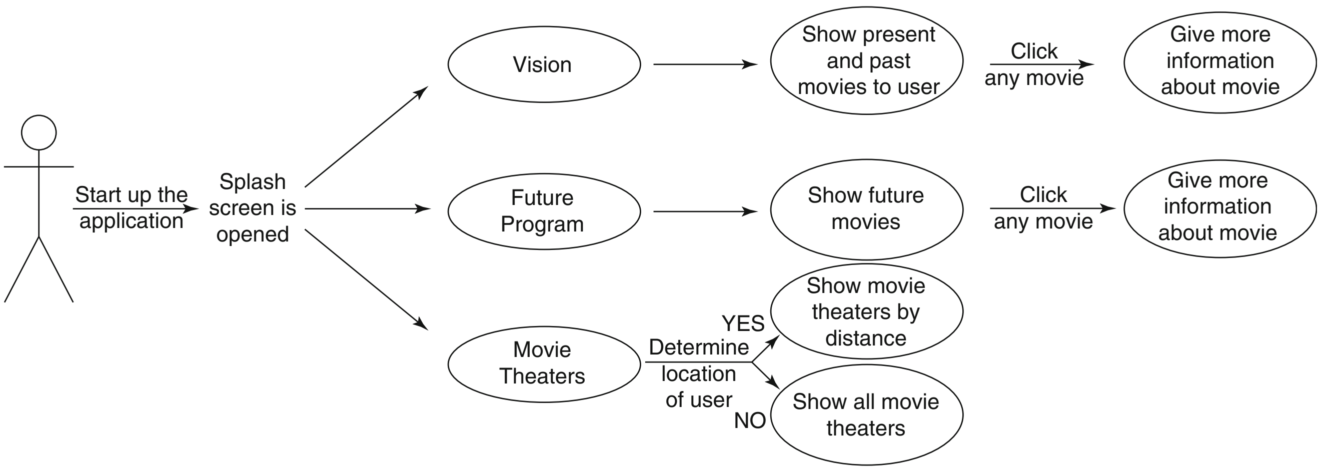
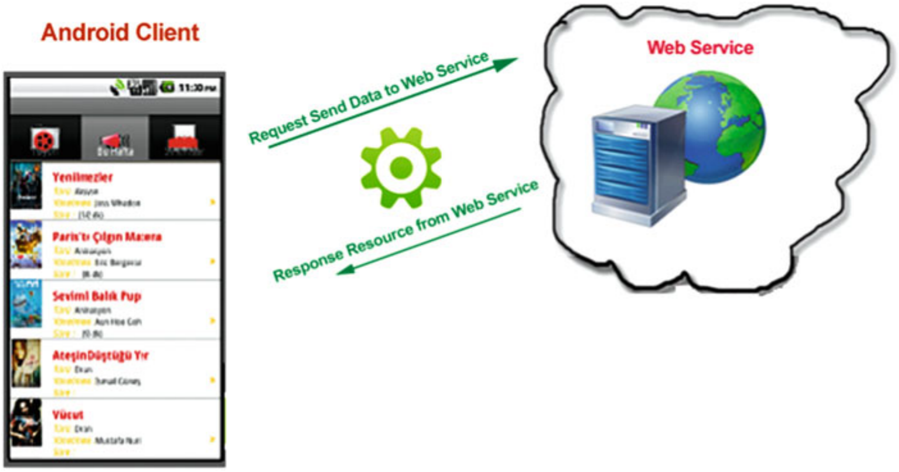

Start up the application
Splash screen is opened
Vision
Show present and past movies to user
Click any movie
Give more information about movie
Future Program
Show future movies
Click any movie
Give more information about movie
Movie Theaters
Determine location of user
YES
Show movie theaters by distance
NO
Show all movie theaters

Fig. 2 Use case diagram of location-based movie advisor

Fig. 3 Vision screen and request-response representation to web service



click to any movie theaters which are very close to the users, they can see vision movies.

Movie advisor applications are preferable from a lot of people in today. There are lots of different types applications are also available in today's market for the same aim. Different from other applications, we create this application not only for the "best" provider. We want to make a difference at sector and do whether before in and is collected all the other properties in the same Android system. There are several reasons why our location movie adviser system should be preferred.

Priority to the User's Requests

System identifies, learns and understands user who has got authority in cinema determination and eliminate the cinema which has never want to go. So user decision is very important, and the software behaves regarding user's decision and request.

User Friendly

The architecture of the system is designed in a noncomplex infrastructure to use comfortably in the application. Screen transitions are so easy to users, so users can find whatever they want to see about movies.

To Be Faster

The system has been designed to ensure that users will not wait more and find information as soon as they expect. Application start-up time, determination of movies lists according to location of the users and movies that users may be curious about are archived easily by proposed architecture.

Changeable

The advantage of this system is that, it is to be developable and replaceable so it could open innovation at the android market.

Having a Wide Range of Information

When user starts using this application, their location is founded according to their area. Then, according to location information of users, system shows them movie theaters which are at shopping centers in Istanbul and seasons which is nearest to them. In addition, user can learn today's program and upcoming movies, movie content, the location of the movie theaters and contact information of the movie theaters and movie trailers.

Web Based

If mobile phone or tablets of the users have android system, downloading and installation the application is so easy. It is not necessary to search this program throughout the Internet or give money to download it so users reach it easily whenever they want.

Implementation

In this section, we have a close look to implementation of Location-based movie advisor application. We explain design steps and give the flow chart of the application. Real Simple Syndication (RSS), Web Service, Photoshop CS 5.5, Eclipse IDE for Java Developers, and Pyhton are used for developing the application of Location-based Movie Advisor [6–10]. Explanation of the Algorithm in details is given below.

Cinephilia is considered as a location-based application and it has got two main parts. First part is a web service which stores longitude-latitude (as a pair) information of movie halls in Istanbul and movie's time table. The other part is client part on an Android mobile device. The client device has to have a powerful Wi-Fi connection because it must communicate with the web services. Thus, the application uses Android's Network Location provider.

First of all the client application listens to location updates only while in the running state to minimize the battery consumption. Because of that, the client connects to the Internet at the beginning of the Android application life cycle. The client gets the location information of the user using Wi-Fi connection and tries to find the closest longitude-latitude pair comparing to that information. In order to perform that, the application gets a list from the web service. However it does not need to search entire list because the application can eliminate the other possibilities using client's location.

Secondly, we get the system date and then check the candidate's movie times. Then Cinephilia finds movies which will be shown in that time. According to screen independent Android layout objects, the application lists the movie information. The client application disconnects the Internet when the user kills the process. If the user never kills the process, our application continues to listen to location updates.

Flow chart of location-based movie advisor application is given in Fig. 4. Two main parts of the application can be seen, and also main steps can be described in the flow chart. Cinephilia is considered as a location-based application and it has got two main parts. First part is a web service which stores longitude-latitude (as a pair) information of movie halls in Istanbul and movie's time table. The other part is client part on an Android mobile device. The client device has to have a powerful Wi-Fi connection because it must communicate with the web services. Thus, the application uses Android's Network Location provider.

Our system is created in a non-complex and easy design by using lots of important technologies at the below.

Parsing the Data

Firstly, the searching of web services is started to parse the data. Web services can convert your application into a web-application, which can publish its function or message to the rest of the world. The basic web services platform is xml plus http. We use two web service pages, these are http://rss.beyazperde.com/filmler/bunlar?format=xml and http://rss.beyazperde.com/filmler/pekyakinda?format=xml. Then we parse these xml pages.

TabHost Layout

Secondly a TabHost Layout is created and then four Activities embedded to four Tabs. Three tabs include a ListView. Vision and Future Program and Saloons pages have Listviews.

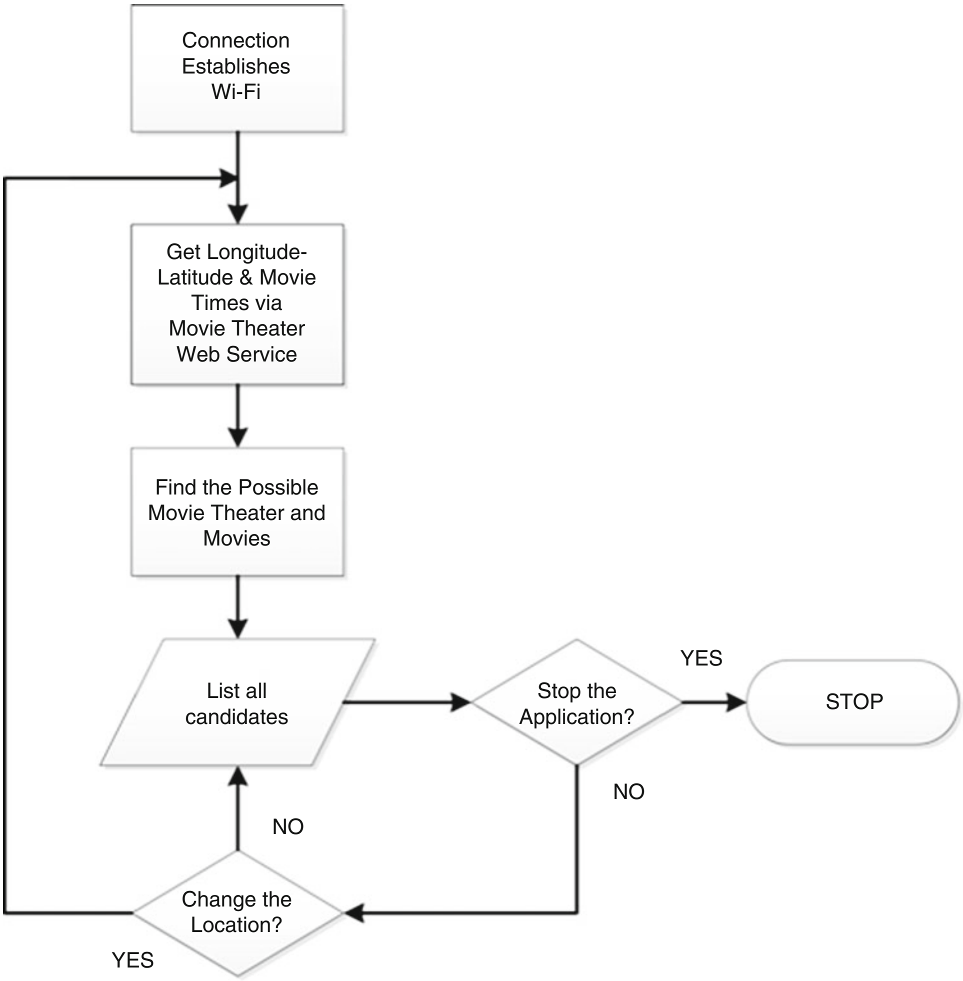

Connection Establishes Wi-Fi
Get Longitude-Latitude & Movie Times via Movie Theater Web Service
Find the Possible Movie Theater and Movies
List all candidates
Stop the Application?
YES
STOP
NO
NO
Change the Location?
YES

Fig. 4 Flow chart of a location-based movie advisor application for android devices

AsyncTask Thread

Three tabs of program have AsyncTask thread. AsyncTask enables proper and easy use of the UI thread. This class allows to perform background operations and to publish results on the UI thread without having to manipulate threads and/or handlers. Our program shows "Loading" message with AsyncTask thread. An asynchronous task is defined by a computation that runs on a background thread and whose result is published on the UI thread. An asynchronous task is defined by 3 generic types, called Params, Progress and Result, and 4 steps, called onPreExecute, doInBackground, onProgressUpdate and onPostExecute.

Custom Adapter

Custom adapter is utilized list to parsing data. BaseAdapter is an array of arbitrary objects. By default this class expects that the provided resource ID references a single TextView. If you want to use a more complex layout, use the constructors that also take a field ID. That field ID should reference a TextView in the larger layout resource. However the TextView is referenced, it will be filled with the toString() of each object in the array. You can add lists or arrays of custom objects. Override the toString() method of your objects to determine what text will be displayed for the item in the list. ImageViews results fill the views, override getView to return type of view.

LayoutInflater

We use LayoutInflater. Instantiates a layout XML file into its corresponding View objects. It is never used directly. Instead, use getLayoutInflater() or getSystemService(String) to retrieve a standard LayoutInflater instance that is already hooked up to the current context and correctly configured for the device you are running on.

RssHandler

RssHandler class is created to parse images. Because we did not parse images in <enclosure url:> tags. So we use DefaultHandler and SaxParser.SaxParser. These define the API that wraps an XMLReader implementation class. However this interface was replaced by the XMLReader, this class wrapped the Parser interface in JAXP 1.0. For ease of transition, this class continues to support the same name and interface as well as supporting new methods. An instance of this class can be obtained from thenewSAXParser() method. Once an instance of this class is obtained, XML can be parsed from a variety of input sources. These input sources are InputStreams, files, URLs, and SAX InputSources. This static method creates a new factory instance based on a system property setting or uses the platform default if no property has been defined. The system property that controls which Factory implementation to create is named "javax.xml.parsers.SAX ParserFactory". This property names a class that is a concrete subclass of this abstract class. If no property is defined, a platform default will be used. As the content is parsed by the underlying parser, methods of the given HandlerBase or the DefaultHandler are called.

Listview

Each Listview's subtitle has an image a title and subtitles. And we use OnClick event for every subtitle of Listview in the Vision and FutureProgramme pages. If users click to Subtitles, they can see movie's details.

GetListfromXML Methods

ListfromXML methods are used to parse Xml data and the data is embedded to empty text views.

DocumentBuilder

It defines the API to obtain DOM Document instances from an XML document. Using this class, an application programmer can obtain a Document from XML. An instance of this class can be obtained from the newDocumentBuilder () method. Once an instance of this class is obtained, XML can be parsed from a variety of input sources. These input sources are InputStreams, files, URLs, and SAX InputSources.

JSON Object

Our customized HTML data to JSON Object are converted. Then we create a JSON service from on Internet for Movie Theaters. We take JSON data from JSON service and we hold these data in a class. We will hold all JSON data in a class. Also JSON data have Saloons' names, Saloons' latitude and longitude variables. Movies' Names and Sessions are included by Movie Theaters.

Location Manager

The aim of the Location Manager is to find current location of the users. When program find the location of the users, it searches all latitude and longitude variables of movie theaters. If the radius of the current location of user covers the radius of the latitude and longitude variables of the movie theater, the movie theater is listed. Finally users can see all movies in the Vision and Future Program, and they can see the nearest movie theaters and movies belonging to these movie theaters and also they can follow movie times from this application.

Complete System Architecture

People usually prefer to go shopping centers. Shopping centers' number is becoming increase day by day so people can come across them anywhere. They usually want to go to the movies in shopping centers after eating and shopping. Cinephilia application is addressed to people who love movies. It includes all shopping center cinema malls' information in Istanbul and provides them with information about their location after their location search. There are several tabs and programs are used in this application, these are vision, this week and movie theaters.

There are several steps for program execution. Firstly, a splash screen will be opened and the user interface comes to it. There are three tabs that vision, future program and cinema halls are seen in the screen while user interface is started to work. If user clicks the cinema halls tab, notification bar comes to screen and asked to user they can be determined user's location. If user accepts it, the movie theaters from near to far are listed to user. In addition, movies and movies times are seen from chosen movie theater. After clicking the vision tab, vision movies and their directors, duration, type, and posters are informed to users. User can get more information about movies when click any movie they wish.

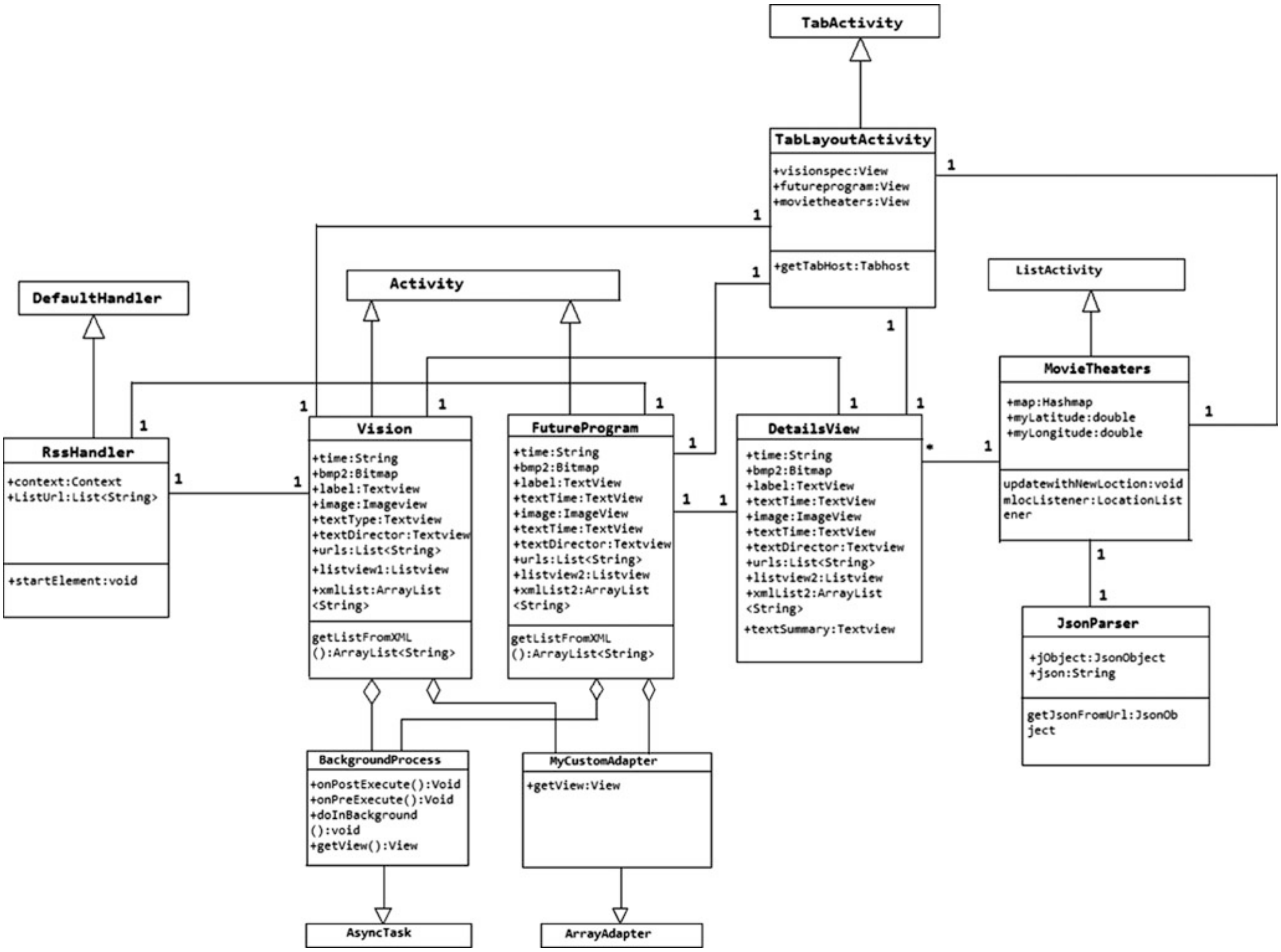


Fig. 5 Software architecture of location-based movie advisor

When future program tab is selected, today's movies and directors, duration, type, and posters are informed to users like vision tab and users learn more information after clicking any movie.

Furthermore, Parsing the data, TabHost Layout, AsyncTask thread, Custom adapter, Listview, LayoutInflater, GetListfromXML methods, DocumentBuilder, JSON object, Location Manager, RssHandler, LayoutInflater are used remarkable Technologies for Cinephilia. Complete software architecture of Location-based Movie Advisor is given in Fig. 5.

Conclusions

Location-based applications are significantly different from any ordinary applications for mobile devices. While a great amount of application has been developed for mobile application, a very few application have been uploaded to mobile markets. Cinephilia is one of the few location-based applications developed for Android devices. Cinephilia firstly determines the location of the users and lists movie theaters to them, so it simplifies the lives of users who go to shopping centers and want to go cinema easily and quickly. Moreover, it shows the most recent movie times so users can go and buy their tickets. In addition, Cinephilia give information to users about movies whatever they want such as director name, movie subject, and movie times, type and cast.

Cinephilia is a simple, non-complex, understandable and attractive. It is designed according to requests of users. While developing Cinephilia, some important design criteria such as user-friendliness, easy to use, and new technologies such as parsing movie information, determination of longitude and latitude of the user are considered.

Briefly, a location-based movie advisor application for Android devices is developed, implemented and explained. We believe that location-based applications will be important part of mobile devices. So, developing this type of application gives a significant contribution to the mobile market. Next generation location-based application development will be potential area for further study of us.

References

1. S. Sheck, *Next-Generation Location-Based Services for Mobile Devices*, CSC, Grant 2010.
2. S. Conder, L. Darcey, *Android Wireless Application Development* Second Edition, December 2010.
3. http://en.wikipedia.org/wiki/Location-based_service.
4. http://developer.android.com/guide/topics/location/strategies.html.
5. Mark L. Murphy, *The Busy Coder's Guide to Android Development*. Commons Ware, LLC, 2008.
6. R. Rogers, J. Lombardo, Z. Mednieks, B. Meike, *Android Application Development*. O'Reilly, 2009.
7. Marko Gargenta, *Learning Android*, O'Reilly, 2011.
8. Jason Morris, *Android User Interface Development Beginner's Guide*. Packt Publishing, 2011.
9. L. Darcey, S. Conder, *Teach Yourself Android Application Development in 24 Hours*. Sams, 2010.
10. S. Hashimi, S. Komatineni, D. MacLean, *Pro Android 2,* Apress, 2010.