**Tallinn University of Technology**

**Business information systems**

**Filip Paunović**

**Adrijan Jakšić**

**Contents**

[1. Introduction 3](#_Toc434758514)

[2. Business model 4](#_Toc434758521)

[2.1 Value of propositions 4](#_Toc434758523)

[2.2 Key activities 5](#_Toc434758524)

[2.3 Key partners 5](#_Toc434758527)

[2.4 Key resources 6](#_Toc434758532)

[2.5 Market segments 7](#_Toc434758537)

[2.6 Channels 7](#_Toc434758551)

[2.7 Value chain structure 8](#_Toc434758556)

[2.8 Competitive strategies 8](#_Toc434758561)

[2.9 Cost structure 9](#_Toc434758565)

[2.10 Revenue generation 10](#_Toc434758567)

[2.11 Business model canvas 10](#_Toc434758569)

[3. Business process description 11](#_Toc434758570)

[4. Access database 12](#_Toc434758572)

[4.1 Database 15](#_Toc434758573)

[4.2 Example of functions we will use 17](#_Toc434758574)

[5. Balanced scorecard 18](#_Toc434758575)

[6. InVision - Prototype 19](#_Toc434758576)

[6.1 The prototype software solution 19](#_Toc434758577)

1. **Introduction**

The goal of this project is the creation of mobile application based on the concept of „Twitter“, but specializing for closed communities such as public institutions (universities, schools, hospitals ...), sport centers (gyms, football associations) and private companies (covers all business areas).

For a form of communication based on one-way communications with brief messages (statuses) there is a universal need. The proof of that need for this concept is Twitter that is globally accepted and today has more than 280 million followers. The advantages of Twitter and the reason for its success is that with no restrictions you can follow any user in the world, and every user can keep track of you, by reading your own brief messages (statuses). This makes it an ideal model of the flow of information in all forms of communities.

The need for such a concept of communication based on one-way short messages exists even in closed communities. The weakness of Twitter in this area, after we conducted research, is that the information is available to any other user outside of that community in the world.

What exactly do we develop? Our application is based on the security and closeness of the community itself and communicating with one-way short messages. Also, the application is primarily designed for mobile devices, because of the extreme popularity of smartphones. Selected are Android & iOS platforms due to the prevalence and accessibility. It works by giving employees the ability to follow other employees and read their short messages. Each employee can post some information such as text, links and images. Messages are being read in sorted news feed that is the initial interface of our application. There is a possibility of implementation various algorithms by which notifications are sorted by relevance (eg. the message from the boss always has to be read). All this will fulfill every aspect of user needs.

At the moment communities and such organizations use various ways to communicate with each other such as dashboards, Facebook groups, emails but our application will make the communication much easier and faster making their process more efficient.

The name of our application will be *“Twitter” for communities – ADDA*. The name is as an acronym of “Add a... person”. The name itself is catchy and explains very well what the main purpose of the application is.

1. **Business model**

What is the state of the market? Currently, there are several solutions adopted for communications within a closed community. The most accepted solution is Microsoft Outlook, which offers the ability to communicate via private e-mail address (eg. Official faculty webmail). There is also „Slack“, which is based on the chat in real time and of course Facebook that allows you to create closed groups.

* 1. **Value of propositions**

Application will be made specifically for mobile devices and just for Android and iOS operating systems. It will have the ability to log on private social network of several different groups, but only one at a time. It works by giving users the ability to follow other users and read their short messages. Each user can post some information such as text, links and images. Depending on the agreed requirements it may have the possibility of private messages, various algorithms for refreshing news feed etc.

Defining problems that our application will solve:

|  |  |
| --- | --- |
| Problem | Network Privacy |
| Includes | Private and secure network |
| As a result, | users are careless |
| The successful solution would be | Set of their own IDs and passwords |

|  |  |
| --- | --- |
| Problem | More than one group in one application |
| Includes | The possibility of one group at the same time |
| As a result | users have "everything in one place" |
| The successful solution would be | specific IDs for groups |

|  |  |
| --- | --- |
| Problem | Additional requirements from users |
| Includes | Gathering, summarizing and realization |
| As a result | users are completely satisfied |
| The successful solution would be | Clear and agreed requirements |

|  |  |
| --- | --- |
| Problem | Communication |
| Includes | Visibility of posts and the ability to write your own |
| As a result | Easy communication among users |
| The successful solution would be | Communication based on Twitter |

* 1. **Key activities**

Firstly we need to start a company and register it. Then we need to a team of people necessary for developing an application based on our idea. For programming we need a team of 2 back-end, 2 Android and 1 iOS programmers for a period of one year, in order to develop prototype and after that a basic product.

Founders of the company need to raise a loan of 30.000 € just so that they will make a prototype in first 6 months. After that we need to show our prototype and try to find a sponsor that will invest in our idea so we could work on it even 1 year more to accomplish basic product. Also we need to find and rent an office where we will have meetings and where we will discuss about our application progress and development process. One day when we will have a basic product we need to advert and present it to our possible buyers.

* 1. **Key partners**

Our main partners are other companies that are helping us in the process of developing our product. One of them is Seapine Software Inc. and they are helping our developing process with their product called “TestTrack RM”. That product helps out team be more efficient and present our progress more transparently so the team is more aware of the process. Also it has the ability that team leader can organize and assign all the duties in the process. We can also assign deadlines for all the duties that are assigned to our team.

There is also another program called “InVision” that helps us present developed prototype of our application. It is a powerful management and presentation tool that will help us in better convincing sponsors to invest in our idea so the basic product can be accomplished.

Next partners are the mentioned investors that will help us in our financial situations before so we could continue to work on our idea of application.

Last partners are the future users (companies and organizations that will buy our product) who will send us feedback through our application about all the problems that our application may have and about all requirements that we need for developing a better and safer application.

* 1. **Key resources**

The main key resource is our team of developers. They are crucial for our idea to come true.

Next resource is the rented office where our meeting will be held and where we will discuss all the matters about our business and development process.

Also, the server where the database will be stored and interact with our user applications. This is a very important resource because we need to have our information stored somewhere on the internet so every user can access it allowing all the functions of our app work correctly and as expected.

However all the mentioned technological and human resources cannot be possible without investors who will give us some foundation for our financial needs.

* 1. **Market segments**

Interested groups, potential buyers and users of our app are:

* Public institutions
  + Universities
  + Schools
  + Hospitals
  + Government, etc.
* Private companies
  + Medium to large (at least 20 employees)
* Sport centers
  + Gyms
  + Sport associations, teams
* All sorts of communities

What is the market size? According to our evaluations, in the world there are over 100,000 different closed communities that would be interested in such a service. Their costs on the various communication mechanisms on average amount to 160,000 Euro per year. Our goal is to win at least 1% of that market, which would allow us to become extremely profitable company.

* 1. **Channels**

At the beginning we will have a lot of presentations of our prototype in the phase of trying to find some investors. After on, when we will have the basic product we will have a lot of presentation to many potential companies and communities that would be interested.

All the information about our product will be shown on our website with detail specifications and offerings. Additionally, we will advert our product throughout the internet and by Google Adwords that will be very useful in advertising our product to the whole world.

We will take our application and present it on a lot of different and major industry fares and trade shows through all around the world.

Each of our future users will also advertise us through their websites and other possible channels.

* 1. **Value chain structure**

ADDA application guarantees high quality service that includes fast response system and very secure network for our users. All the requirements gathered by our service will be professionally achieved with high performance.

The app has implemented the option of giving information about any problem or suggestion about the current version. The head of the community can then send us feedback through the application about those problems & suggestions. We will happily and professionally reply and try to solve all the problems and make our app better and safer. Also in the phase of maintains the app has a very easy way to communicate with our team.

We offer the basic package for the same price for every community. Additional options and abilities of the app are also possible to make but that package will be customized for each community so the price will be also different. If some of our current users wants to buy additional package we will have promotional price for them as an old customer.

The quality of our product will be non-negotiable and we will always try to make the best effort for highest performance, security and privacy.

* 1. **Competitive strategies**

First advantage of ADDA is the model of communication where with no restrictions you can follow any user and every user can keep track of you, by reading your own brief messages (statuses). This makes it an ideal model of the flow of information in all forms of communities.

Application will be made specifically for mobile devices and just for Android and iOS operating systems. It will have the ability to log on private social network of several different groups (“everything at one place”), but only one at a time.

On top of all of the described advantages of the concept, we offer the community the ability to adapt the application with their personal preferences of the community. We are going to do that by collecting feedback from users after the first deployment of the beta version of our application. These adjustments will be ​​various changes such as visual adjustment (adjustment of the visual identity), functional adjustment (add features private messaging - inbox, prioritizing posts in the news feed by the hierarchy of employees). These adjustments give employees the possibility to personalize the system and meeting all their needs. Adjustments would then be grouped in packages that would be additionally charged.

* 1. **Cost structure**

Our activities include the programming and marketing of the product itself. For programming we need a team of 3 back-end, 2 Android programmers and 2 iOS programmers for a period of one year, in order to develop a basic product. The cost that we must pay for our team is calculated by adding all the salaries of the members.

We have a rented office where we have all the meetings with the team members, sometimes with the partners that we need to pay every month. Without the office our idea can’t come true, because some things we must agree in person. Also it is a good thing to have a place where potential buyers can visit us and even just for team building, because that is also important for team cohesion.

Next thing we need money for is to buy professional tools, like TestTrack RM or InVision that will help us in the development process to be more efficient and transparent to our buyers and partners.

Also we have expenses that includes some business trips to our potential buyers for presentation of our product or to some industry fares and trade shows.

The last thing we should need money is for advertising our app on internet, for example. Google Adwords.

* 1. **Revenue generation**

Marketing activities would be carried out by advertising on the Internet (Google Adwords) and direct presentations in the institutions themselves.

When meeting potential buyers we will present them our basic product that will be included in the basic package with the lowest price. Also we will let them know of some other additional options that our product can have if they want it. But of course for additional abilities we will charge them higher. If the potential buyers have some special requirements for our product we can also make a deal with them about the cost.

After the agreement about the package they want to buy and even after they start to use the product they can send us feedback and message us about wanting an upgrade of abilities through the app.

* 1. **Business model canvas**

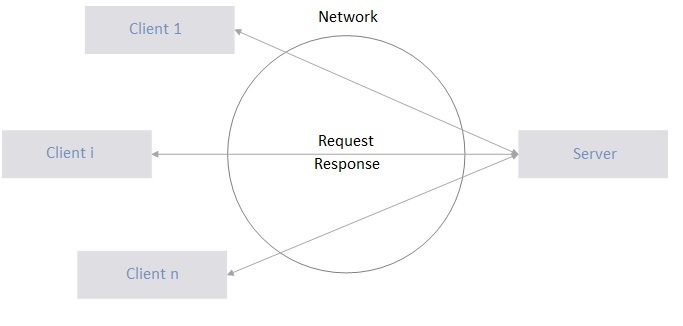
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Value proposition** | **Key activities** | **Key partners** | **Key resources** | **Market segments** |
|  |  |  |  |  |
| **Channels** | **Value chain structure** | **Competitive strategies** | **Cost structure** | **Revenue generation** |
|  |  |  |  |  |

1. **Business process description**

This principle will be achieved by making each community getting its installation package defined by internal id (e-mail of the facility). Thus, the user can, via a single installed application, login to the system of any community that uses our application and of which he is a member, and it is checked by specific combination of e-mail (username) and password.

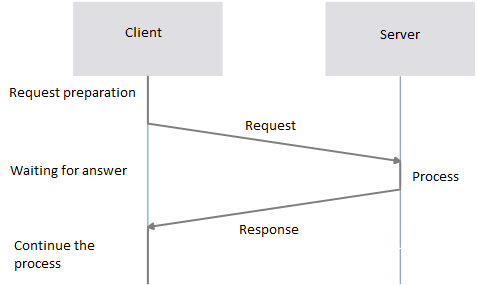
1. **Access database**

The system was realized using the client - server model. The server consists of a database and access point for communication with the mobile application. The access point is based on REST form using the HTTP protocol to communicate with the client. The contents of HTTP messages is shown in JSON format.



**Figure 10 Client-server model**

As shown in Figure 10, the client is requesting service from the server, and the server provides it for multiple clients with the property that each user has the impression that the system provides service only to him.  
Communication with client and server is starting with the client's request for service and waiting for replies. The server accepts and processes the request and returns the response to the client. The client and server communicate by exchanging requests and responses as shown in Figure 11. So, the functionality of the client – server is in a greater focus than its performances.



**Figure 11 Communication between client and server**

The interaction of client and server involves the preparation and transmission of the request and waiting for server to reply. The server processes the received request and returns a response to the client, which is then further processed.  
In the model client - server all the clients use the same interface on the server that defines a set of requirements that can be set. Server functionality is defined in advance during the time of system design: the server must have all the means necessary to serve all the requests. Software implementation of the client – server system includes the following:

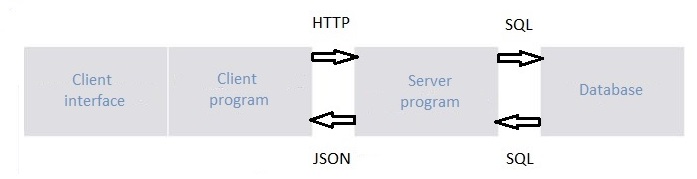
• User interface

• The client application

• The server application

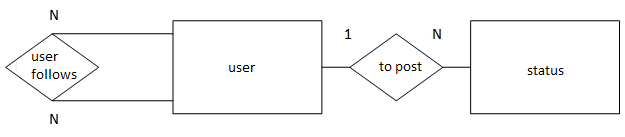
• Database

The server is additionally being performed with three separate computers and then it is called a three-tier physical architecture. In this model server program acts as a server to the client and as a client to the database. Client program sends REST interface requests via http protocol and receives responses in JSON format. Requests are coming to server app that sends SQL queries to the database which sends back the desired information. This model is visible in Figure 12.



**Figure 12 Three-tier architecture**

The database is the central location of all data in the application. For it to work correctly, it is necessary to ensure data integrity and design with focus on the normal forms. Under the integrity of a database we mean correctness of the information contained in the database. In a wider sense, the integrity of the database covers all measures intended to prevent the entry of incorrect data in the database. Therefore, on databases are applied restrictions representing limits of the database contents on allowed conditions that ensure consistency (mutual compatibility of data) of database when entering, updating and deleting data. Restrictions are often the result of semantic properties of certain entities, attributes and relationships, and as such must be built-in the appropriate database model when designing the database. When structuring a relational schema in a relational database, the goal is to achieve such a structure that will prevent the appearance of undesirable properties in the relations. Properties of relations depends on the semantic relationships between attributes in a relational schema. By limiting these relations to particular types of dependences can be achieved that the relations in the relational schema do not have undesirable properties. For a relational schema in which such restrictions apply is said to be normalized, that is in normal form. One simple ER model of basic case uses through the database is shown in Figure 13.



**Figure 13 ER model**

Specificity of REST interface for communication between client and server is that it puts the emphasis on resources and the way they are addressed and transmitted via HTTP. Because of its extreme simplicity of use, it is used above SOAP and WDSL principles. REST ensures that the client and server are completely independent of the internal work of another.  
To exchange data we will be using JSON format. It is very easy to read and write and is also simple to parse and generate. It can be used from any programming language, but it is especially close to C and his family of languages ​​(C ++, Java, Python, JavaScript ...).  
The server application processes requests requirements, manages clients, provides security and consistency of the database.  
For communication performance of Android application and server via HTTP protocol is required proper library on each side. The client-side (KPP) is required to have libraries to handle the HTTP messages. Due to the large and flexible programming interface and stable implementation of the best choice is the Apache HTTP library. It contains classes that are supporting basic elements of communication via HTTP, such as abstraction of messages with headers and accessories and abstraction of communication through which these messages are exchanged. The user interface is all that the user sees when the application is running. Through this module the customer specifies all their demands and sees responds. To application maintain user interface active and responsive, it is necessary to run user interface and communication with the server in separate threads.

* 1. **Database**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Purpose** |
| User\_id | int | User unique identifier and the primary key; |
| nickname | char | The name under which the user is represented to other users; |
| password | char | User authentication; |

The relation 1 - user

|  |  |  |
| --- | --- | --- |
| Attribute | Data Type | Purpose |
| Post\_id | int | Unique identifier of post and primary key of relation; |
| User\_id | int | Unique identifier of author of the post |
| time | Date\_time | Time of posting the post |
| message | char | Text part of the post |
| image | char | Image part of the post |

Relation 2. - New post

|  |  |  |
| --- | --- | --- |
| Attribute | Data Type | Purpose |
| Id\_user\_following | int | Foreign key that indicates user who is following other users |
| Id\_user\_followed | int | Foreign key that indicates tracked users |

Relation 3. – User following

* 1. **Example of functions we will use**

|  |  |
| --- | --- |
| Function | Purpose |
| register\_user () | Register user in the system. The user sends nickname and password to the server and gets a reply with a unique user number if the nickname does not exist or „False“ if there is a same nickname already in the system; |
| authenticate\_user () | User authentication. The user sends a nickname and password to the server and gets a reply with a unique user number if the nickname and passwords match or “False” if there is no match; |
| add\_post () | Adding a new text post. The user sends the post to the server and gets a unique number of the post if adding succeeds or “False” value if there is some error; |
| add\_picture () | Adding a new picture in the post. The user sends the post to the server and get a unique number post if adding succeeds or “False” value if there is some error; |
| get\_feed () | Retrieving all the visible posts |
| get\_picture\_from\_post\_id () | Retrieving images from the server. User gets image of a post that had previously received in get\_feed. |

Relation 4. – Function examples

1. **Balanced scorecard**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Theme: Operating Efficiency** | **Objectives** | **Measures** | **Targets** | **Initiatives** |
| **Financial Perspective** |  |  |  |  |
| **Customers Perspective** |  |  |  |  |
| **Internal Perspective** |  |  |  |  |
| **Learning Perspective** |  |  |  |  |

1. **InVision - Prototype**

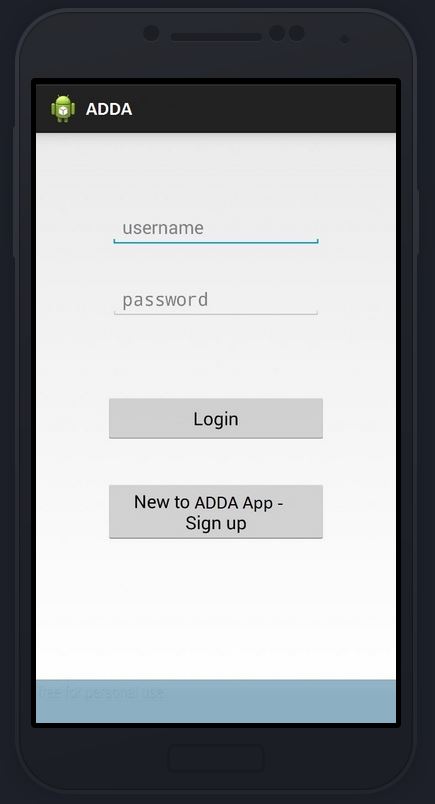
InVision is, in our opinion, proved to be very useful and intuitive tool that assists in the preparation of prototypes and the distribution of the same. It's very simple, developers who have created it thought that everything must be quite intuitive, and to have more one-click options to speed up the work with him, especially when you get to know the application. There are surprisingly many tutorial videos for prototyping applications and their interactivity in InVision. These tutorials are brief and clear, thoroughly and are separated and addressed properly. For us, the only problem in those videos is that sometimes person is clicking too fast and showing stuff. Then the video has to be stopped in these places a couple of times to see exactly what was done.

The tool has 4 modes in the development of a prototype, the first *preview mode* that where we test how the app works, another *build mode* in which we build functionality of the buttons and connect various activities, the third *comment mode* in which we can put some comments if for example some feedback. The last mode is the *history mode* in which we can see the previous versions of the applications that we have saved. The app can be easily shared with colleagues online. As part of this project, tool has been extremely useful for prototyping software solutions which is discussed further in the next section.

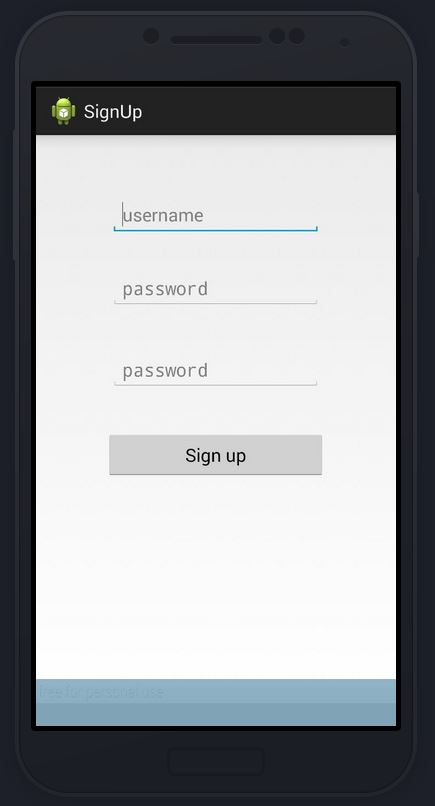
* 1. **The prototype software solution**

Selected are some basic use cases and in order are: user registration, user login, reading news feeds, like and share pictures and messages, start and stop tracking users.

Figure 14 is the initial screen for registration. By clicking the button "New to ADDA App - Sign up" with the flip right option you go to another activity shown in the image 15 where is the addition of new users who do not yet have an account. Figure 14 shows the initial activity where we enter a username and password and press the button Login to enter into the application and to open the initial news feed shown in image 16th

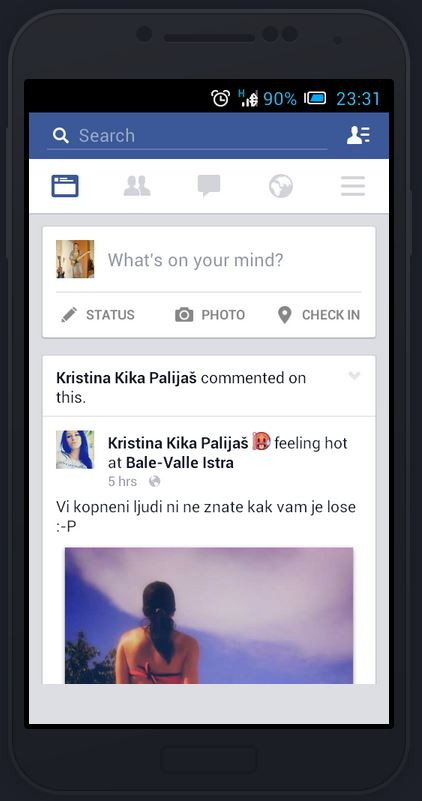


**Figure 14 Home screen**



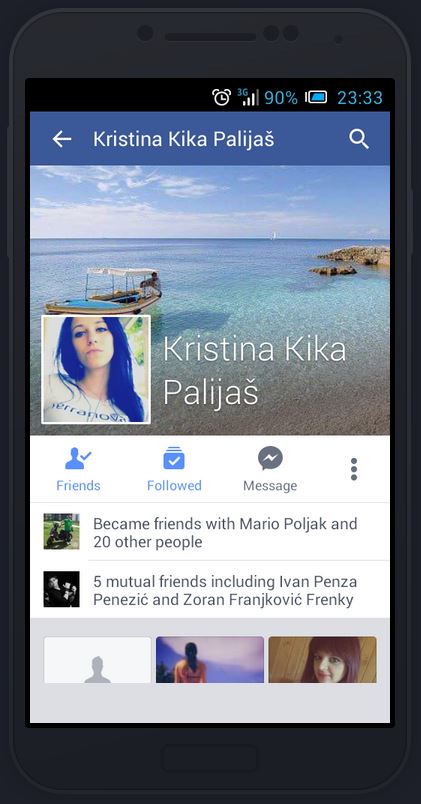
**Photo 15 Sign up actions**

In this window you can add new users, that is, creating accounts for the application. Pressing the button "Sign up" we return with flip left action to the home screen with image 14.



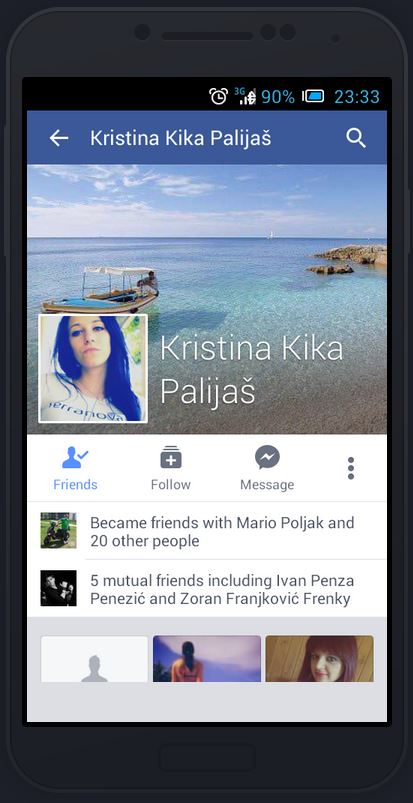
**Figure 16 Initial news feed**

In the tool InVision on this activity we made four functional buttons. Clicking on a picture of my profile we enter in activity „my profile“ shown in Figure 22nd. Clicking on the profile picture or name Kristina Kike we can enter in her profile in the application which is diplayed in picture 17. Clicking on the photo, which is published on the news feed, we enter in the image preview that is displayed in the picture nr 19. Various display options were used in transition from some activity to another activity using some common sense and logic.



**Figure 17 Kika profile**

              Clicking on button "Followed" I stop following the user Kika and her statuses (next figure). Clicking on the upper left corner we are going back to the news feed, shown in figure 16.



**Figure 18 Kika profile – unfollowed**

In this activity it can be clicked again to „follow“ button to follow Kika again, so that I can follow hers interesting statuses. Clicking on the upper left corner we switch to the new news feed (because we no longer follow Kika). It is shown on figure 21.



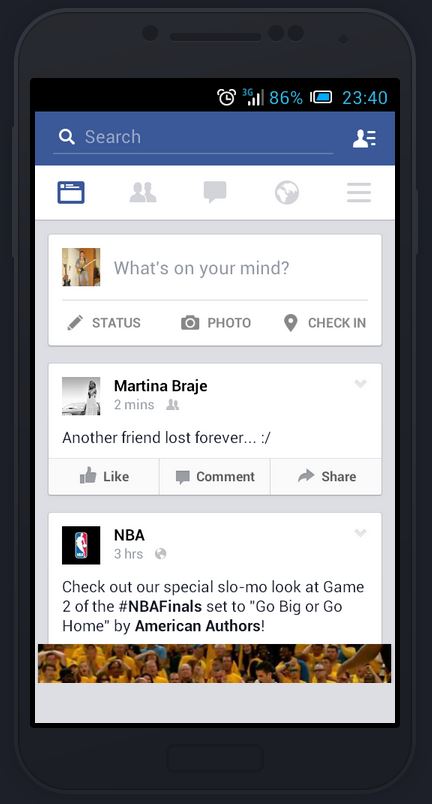
**Figure 19 Kika photos**

              In this photo I made the option of clicking the "Like" to let my friend know that I like these photos. Clicking on the black area of the screen we return back to the news feed with image 16th



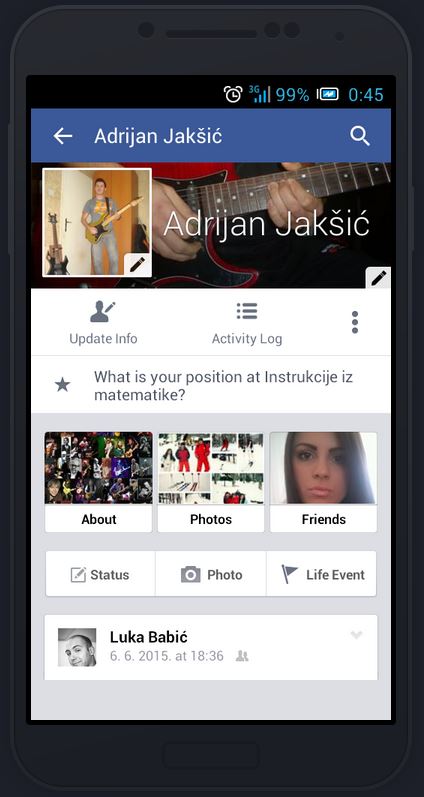
**Figure 20 Kika photo - like**

              This activity, despite low probabilities ☺, has the option of clicking the "Like" to cancel out the effect from the last activity. Clicking on the black section of the screen we transfer to the new news feed (since we liked this picture, there is no need to go back to the old) from the image 21st



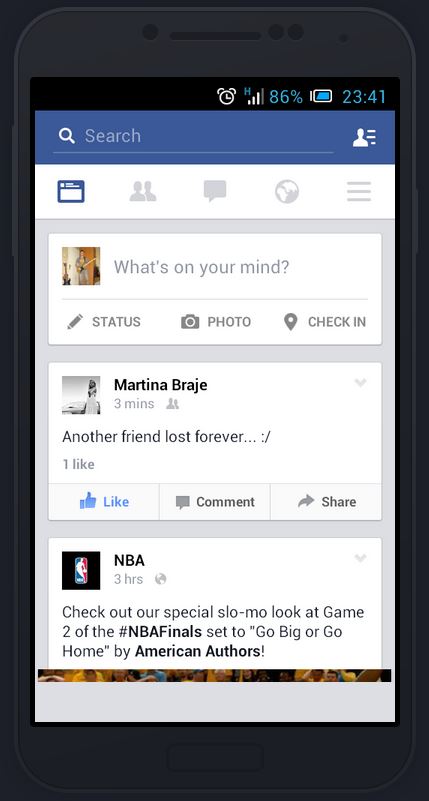
**Figure 21 News feed 2**

              In this news feed, you can click the picture of my profile which opens my profile (Figure 22). By clicking on „Like“ we like some status (Figure 23), and by clicking „Share“, we share the status on my profile (Figure 24)



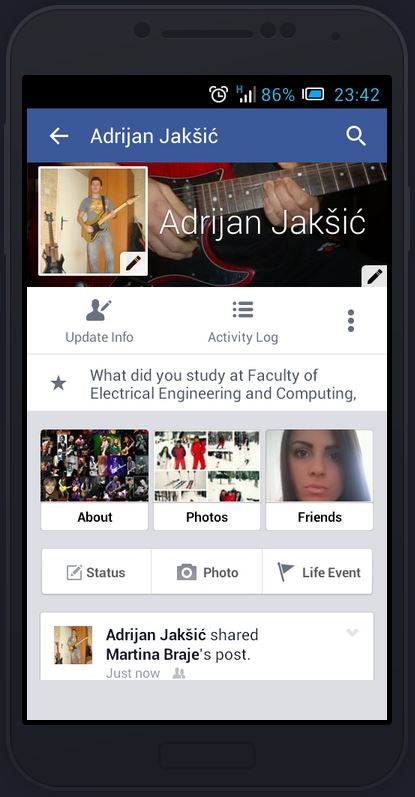
**Figure 22 My Profile**

              By clicking on the upper left corner we return to the figure 22.



**Figure 23 Martina status like**

              This activity has all the same functionality as the one from the image 21.



**Figure 24 Martina status share**

              This activity has only one function and that is to click on the upper left corner which returns to the initial news feed (Figure 16).

This prototype is a basic for the application ADDA, and later we would develop it in the direction of additional features like inbox, report the users and their statuses (for being vulgar or something like that), various algorithms of refreshing news feed etc.