

# **Twitter card: exploring the feature and adding new services**

Konstantin Fedorov

University of Tampere  
School of Information Sciences  
Computer Science / Int. Technology  
M.Sc. thesis  
Supervisor: Päivi Majaranta  
June 2015

University of Tampere

School of Information Sciences

Computer Science / Interactive Technology

Konstantin Fedorov: Twitter card: exploring the feature and adding new services

M.Sc. thesis, 36 pages, 3 index and 24 appendix pages

June 2015

---

This study examines the Twitter card, a feature that adds a preview for rich content links included in tweets. Previous research has not cover this recently added feature, though it has an obvious impact on the user interaction with his/her timeline and provides a new way of representing the user interface. At the same time, a default Twitter client has a major drawback – the number of web services that are officially supported is quite small. This study explores the possible future of Twitter card and presents a prototype application that shows what will happen if all the links will have previews, how this can be implemented and what users think about it.

The prototype application was developed using HTML/CSS/JavaScript, PhoneGap framework and ported to Android operating system. The source code of the application is provided as part of this research.

This study provides also user experience evaluation of Twitter card and the prototype application, and finds out the improvements in the hedonistic perception of the service if the number of supported services increases.

Key words and terms: Twitter card, Twitter, social networks, user interface, rich media

## Contents

1.	Introduction .....	1
2.	Twitter .....	1
2.1.	Description.....	1
2.2.	Categories of messages.....	2
2.3.	Twitter features .....	3
2.4.	Twitter card .....	4
3.	Content rich user interfaces.....	6
3.1.	Introduction.....	6
3.2.	Twitter.....	6
3.3.	Instagram user interface.....	8
3.4.	Digg .....	9
3.5.	Flipboard.....	10
3.6.	Paper .....	11
3.7.	Prototype application .....	12
4.	Preliminary small scale group interview.....	12
4.1.	Method.....	12
4.2.	Implementation .....	12
5.	Application development .....	14
5.1.	Introduction to the system .....	14
5.2.	PhoneGap development.....	15
5.3.	Twitter API connection .....	16
5.4.	Twitter authorization and OAuth.....	17
5.5.	Data storage .....	18
5.6.	Embedding media .....	18
5.7.	HTML and CSS libraries .....	19
5.7.1.	Backbone.js .....	19
5.7.2.	Twitter Bootstrap.....	19
5.7.3.	jQuery.....	20
5.7.4.	twig.js.....	20
5.8.	AJAX loading indicator.....	20
5.9.	Twitter API libraries .....	20

5.9.1.	Codebird.js .....	20
5.9.2.	jStorage .....	21
5.9.3.	twitter-text.js .....	21
6.	Prototype user interface .....	22
6.1.	Authorization process .....	22
6.2.	Main interface .....	24
7.	User testing .....	25
7.1.	Methods .....	25
7.2.	Results of think aloud method .....	26
7.3.	Results of questionnaire.....	27
8.	Discussion .....	30
	References .....	32
	Appendix 1. Background questionnaire.....	37
	Appendix 2. Consent form .....	40
	Appendix 3. Test plan .....	41
	Appendix 4. Summary transcription of user testing .....	43
	Appendix 5. Source code .....	47

## 1. Introduction

In recent years there has been an increasing interest in social networks and particularly Twitter. The idea of writing a message just in 140 characters opened a new way to share information among people.

In 2012 Twitter revealed a new desktop feature called “Expanded tweet” (Sippey, 2012), later “Twitter card”, which shows interactive previews for the links in messages. Users can watch for example a YouTube video directly from Twitter or read the introduction of a newspaper article without opening a new window in the browser.

However, the small number of officially supported web services does not give the opportunity to use these features fully. Although the list of services is not available publicly, some major services are not supported, for example Instagram. Moreover, the Twitter mobile application has even fewer web sites with previews.

This study suggests that modern mobile devices have room for improvement and can support the generation of previews for all the links using automated services. Furthermore, it also illustrates that users appreciate this opportunity. The developed prototype application based on Twitter attempts to address this issue and to prove the idea by conducting user testing based on it. The exploratory user testing reveals the attitude of the participants to expanded tweet feature and to the application itself. The data yielded here can be used for further improving the Twitter user interface, for sharing interactive previews of interesting links.

This study briefly reviews in the following sections the related literature, describes the prototype application structure and development process and results of preliminary, as well as main user study of this application and related questions.

## 2. Twitter

### 2.1. Description

Twitter is a social network that allows registered users to communicate with each other using 140-character text messages, called "tweets" and to unregistered only to read them (Wikipedia, Twitter, 2014). There are different ways to access Twitter – through web, through mobile applications, through SMS, and the like. This network is one of the 15 most-visited websites (Alexa, 2014) with 500 million users (Lunden, 2012).

*Microblogging is a broadcast medium that exists in the form of blogging. A microblog differs from a traditional blog in that its content is typically smaller in both actual and aggregated file size. Microblogs "allow users to exchange small elements of content such as short sentences, individual images, or video links". These small messages are sometimes called microposts (Wikipedia, Microblogging, 2014).*

Twitter became famous during the “South by Southwest” conference in 2007, when the service was presented to public. Online feed was projected to the wall and visitors could exchange their impressions during the event. Next year after that Twitter already has had 6 million users, few thousands of registrations per day and a 600% of growth. In 2009 the growth already reached about 18000% (Akcora and Demirbas, 2010).

Twitter is used for many different purposes; for example, it is an important event backchannel. It has been used to describe important happenings (San Diego fires, Sichuan earthquake, Mumbai terrorist attacks) or as revolutionary machine for controlling street demonstrations in Iran after the presidential elections of 2009 (Rogers, 2013). It has also shown potential for measuring and predicting major depression in individuals - based on messages of people with clinical depression (De Choudhury, 2013).

In addition, Twitter is actively used in marketing and branding: 19% of tweets contain mention of a brand. Nearly 20% of the branding microblogs contained some expression of brand sentiments, from which more than 50% were positive and 33% were critical of the company or product (Jansen, 2009).

However, tweets still can lack credibility, for example when comparing newspaper article and tweet with the same news. Respondents found newspaper articles more credible than microblog messages (Schmierbach and Oeldorf-Hirsch, 2012).

Twitter is mainly English speaking resource. Language activity varies depending on time zone, English tweets being the most popular, followed by Japanese and Spanish. English language versions were remarkably frequent (68%) in two time periods: 2pm and 6pm (Honeycutt and Herring, 2009).

## **2.2. Categories of messages**

Motivations for blogging in general among users can be divided in a few categories (Nardi, 2014): documenting life (recording activities and events), commenting (expressing opinions), getting catharsis (outlet for thoughts and feelings), musing (thinking by writing) and using as community forum (sharing experiences). This is similar to a study (Naaman, Boase and Lai, 2010) that divided all Twitter messages into categories. Human coding and quantitative analysis were applied to the categories of the messages posted by users (Table 1): information sharing, self-promotion, opinions/complaints, statements and random thoughts, me now, question to followers, presence maintenance, anecdote (me) and anecdote (others).

Code	Example(s)
Information Sharing(IS)	“15 Uses of WordPress<URL REMOVED>”
Self-Promotion (SP)	“Check out my blog I updated 2day 2 learn abt tuna! <URL REMOVED>”
Opinions/Complaints (OC)	“Illmatic = greatest rap album ever”
Statements and Random Thoughts(RT)	“The sky is blue in the winter here” ”I miss New York but I love LA...”
Me now (ME)	“tired and upset” “just enjoyed speeding around my lawn on my John Deere. Hehe :)”
Question to followers (QF)	“what should my video be about?”
Presence Maintenance (PM)	“i’m backkkk!” “gudmorning twits”
Anecdote (me) (AM)	“oh yes, I won an electric steamboat machine and a steam iron at the block party lucky draw this morning!”
Anecdote (others) (AO)	“Most surprised <user> dragging himself up pre 7am to ride his bike!”

**Table 1. Categories of messages in Twitter (Naaman, Boase and Lai, 2010)**

Females and users of the mobile application were more likely to post “me now” messages. 20% of users had 53% of messages in “Information sharing” category, having more friends, followers and mentions, than 80% of users having 48% of messages in “Me now”.

Respondents in the study by Counts and Fisher (2011) found 15% of Tweets highly interesting, looking at them for longer time than at other tweets. They would like to click 52% of links in Twitter timeline, but would reply just to 24% and retweet 34% of all messages. A user spent just 3 seconds on average to read a tweet.

### 2.3. Twitter features

The main features of Twitter include the following: hashtags, favorites, mentions, and retweets. A hashtag is any word or phrase immediately preceded by the # symbol. Tweets containing the same keyword or topic will be shown on click on the hashtag. Favoriting a Tweet indicates that the user liked a specific Tweet or put it to bookmarks. All favorited Tweets are saved in Favorites link in user’s page. The @ sign is used to call out usernames in Tweets: "Hello @twitter!" People will use @username to mention a user in Tweets, send him a message or link to his profile. A tweet that the user forwards to his followers is known as a Retweet. It is often used to pass along news or other valuable discoveries on Twitter and always retain original attribution. Up to now,

the research has tended to focus on mentions, retweets and favorites (Twitter, The Twitter Glossary, 2014).

Meier, Elswiler and Wilson (2014) listed the purpose of favoriting tweet as follows: liking tweet (same feature as in Facebook), showing that the tweet is informational, marking the author as a special person (for example, celebrity), and emphasizing that tweet personally relates to the user.

It was shown (Honeycutt and Herring, 2009) that, despite of environment that was not created initially for user-to-user conversations, the @ symbol is a marker of addressivity. The user indicates an intended addressee by typing the person's name at the beginning of an utterance. Thus Twitter also becomes usable for collaboration and recommendations. Most conversations seem to be three to five messages sent over a period of 15 to 30 minutes. 91% of the @ symbol's usage relates to direct tweet to some person, 5% are references and locational "at" took just 0.75%. 33.44% of all tweets start with the "@" symbol (replies) according to the statistics collected during the year 2009 (HubSpot, 2009).

Retweet is just 1.44% of whole tweet population (HubSpot, 2009). This feature is used mainly for the following motivations: spreading information to new audiences, entertaining specific audience, commenting by retweeting and adding new text to it, making presence of another person more visible and so on (Boyd, Golder and Lotan, 2010). Retweets happen more often if the original tweet contains photo, video, quote, number or hashtag (Rogers, 2014).

#### **2.4. Twitter card**

Initially Twitter card was called an expanded tweet and described (Sippey, 2012) as a tweet that has a link to partner websites, which are supported by Twitter to show content previews (showing images, playing videos and more). It can for example be a video from YouTube, The Wall Street Journal article and other rich content. The user can continue to read the article, follow these accounts, and reply to, favorite or retweet the tweet as usual. Later, the expanded tweet was named as Twitter card in the documentation for developers (Twitter, Twitter cards, 2014).

On the basis of the evidence currently available, it seems fair to suggest that Twitter card functionality was actually created to solve partly the problem of information overload (Wikipedia, Information overload, 2014), caused by social networks in the Internet. Exploring information and new data has become increasingly difficult as the volume and diversity grows (Shneiderman and Plaisant, 2005, p.43). A question of which tweet will attract attention of the user was raised by researchers with the increase in the amount of textual data generated by Twitter users (Chun, et al., 2012).



In recent years, researchers have investigated a variety of approaches reducing overload when searching for relevant information in Twitter but nearly all studies have concentrated on automatic methods. Chen et al. (2012) suggested using collaborative ranking to capture personal interests. Alonso, Marshall and Najork (2012) investigated what the actual strategy is for a user to find interesting tweets and proposed a method to score tweets through textual and contextual features.

On the other hand, there is overwhelming evidence of human's natural ability for fast recognition of visual information. Users are working faster with visual rather than with textual representation of information. Software designers are developing appealing user interfaces for input, search and output to manage rich data user interfaces, which can be extended by sound, animation and video (Shneiderman and Plaisant, 2005, pp. 6-8). Moreover, user interface shortcuts for favorite features decrease the amount of interactions and increases their pace. Frequent knowledgeable users especially appreciate this fact (Shneiderman and Plaisant, 2005, p.68).

The eye tracking study by Counts and Fisher (2011) showed that on average the participants spend less time looking at a tweet with links and rate them as less interesting. It also gives weight to the claim that tweets with links do not attract the user more than tweets without links.

While Twitter card requires a link inside of tweet, just 22% of the messages in Twitter contain a link (Crawford, 2014). The evidence from it suggests that expanded tweet will not work for significant part of messages. One possible implication of this is that we cannot just cut out these tweets from general timeline in our research; otherwise we almost certainly will remove some tweets, which are important for user.

The current literature lacks studies about Twitter card, though this feature brought a significant change in the Twitter user interface. Insufficient research on this subject does not allow drawing any firm conclusions about this change. This study continues the Twitter idea of changing user interface and displaying the content preview rather than textual data and extends the amount of supported services and reducing the amount of textual information. A new user interface is presented by the working application based on user's Twitter timeline.

The individual preferences vary by person or task, just as car models or flavors vary (Shneiderman and Plaisant, 2005, pp.24-29). The presented new user interface does not substitute the current Twitter design; however users might find it useful and attractive. Moreover this study reveals how users perceive the Twitter card and this question has not been researched enough in previous studies.

### **3. Content rich user interfaces**

#### **3.1. Introduction**

The web design trends in 2014 will remain in memory with following features: flat design, design firstly oriented for mobile phones and only then to other devices, less text and more content, minimalist navigation, one page design, bigger and better imagery (Lake, 2014).

Similarly, one of the ways to increase enthusiasm among users of system is to use direct manipulation with objects of the system (Shneiderman and Plaisant, 2005, pp.214-215) and giving feeling of involvement directly with the world of objects.

In contrast to textual descriptors, “dealing with representations of objects may be more “natural” and closer to innate human capabilities: action and visual skills emerged well before language in human evolution”. In addition, people work with actions more quickly when those people are given visual rather than textual data (Shneiderman and Plaisant, 2005, p.235).

Modern web services, including Twitter, can have rich content in addition to the textual data. It includes animation (dynamic images), video and audio (clip embedded within a text, graphical document or web page) and different interactions (Dix et. al., 2004, pp.749-756). The rich content user interfaces can be best described with an example of the mobile applications of Twitter, Instagram, Digg and FlipBoard.

#### **3.2. Twitter**

The traditional user interface of a tweet is shown in Figure 1. Twitter mobile application user interface (Figure 2) consists of a table with latest tweets, sorted by date in descending order and it can be scrolled till the last user’s tweet. This table is called “timeline” and shows usually all the latest messages from the accounts that the user is following. Each tweet consists of the user’s thumbnail, his name and username, text of message and a preview, if such is available. Previews can be shown or played by clicking on it. User can reply to tweet, favorite tweet, share it with user’s followers by clicking Retweet button. The details of user’s account can be viewed by clicking on his thumbnail, and then timeline will show only tweets that were written or retweeted by him or her.



Figure 1. Twitter display requirements (Twitter, Display Requirements, 2013)

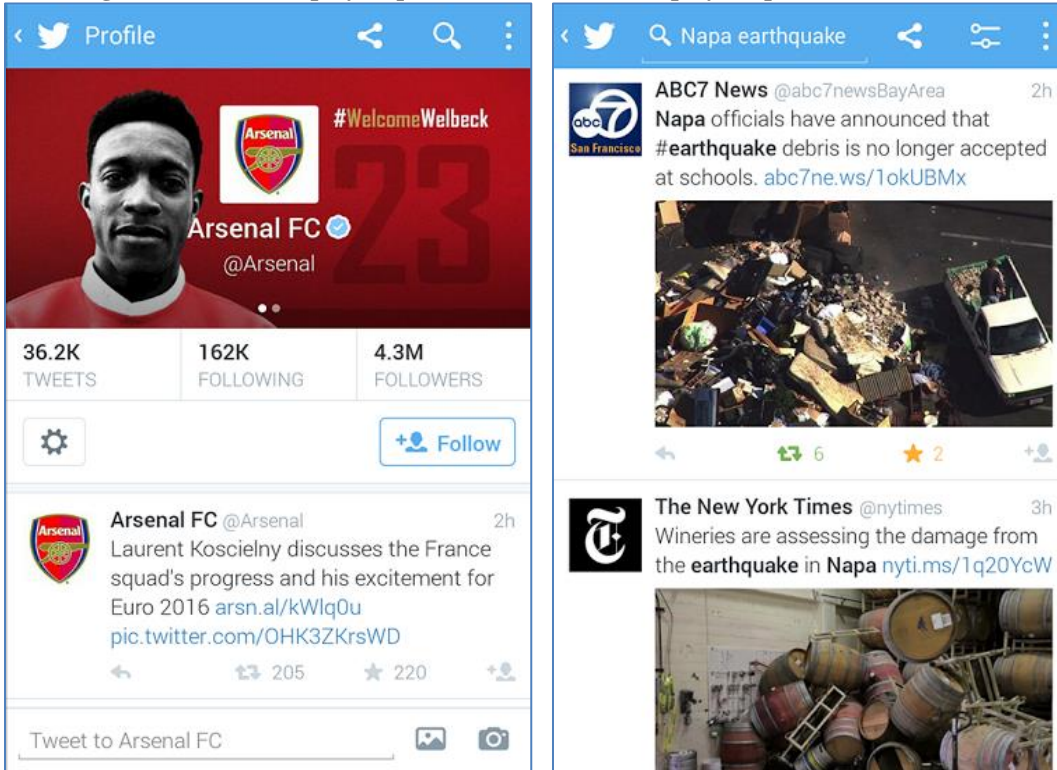


Figure 2. Twitter mobile application (Twitter, Google, 2014)

### 3.3. Instagram user interface

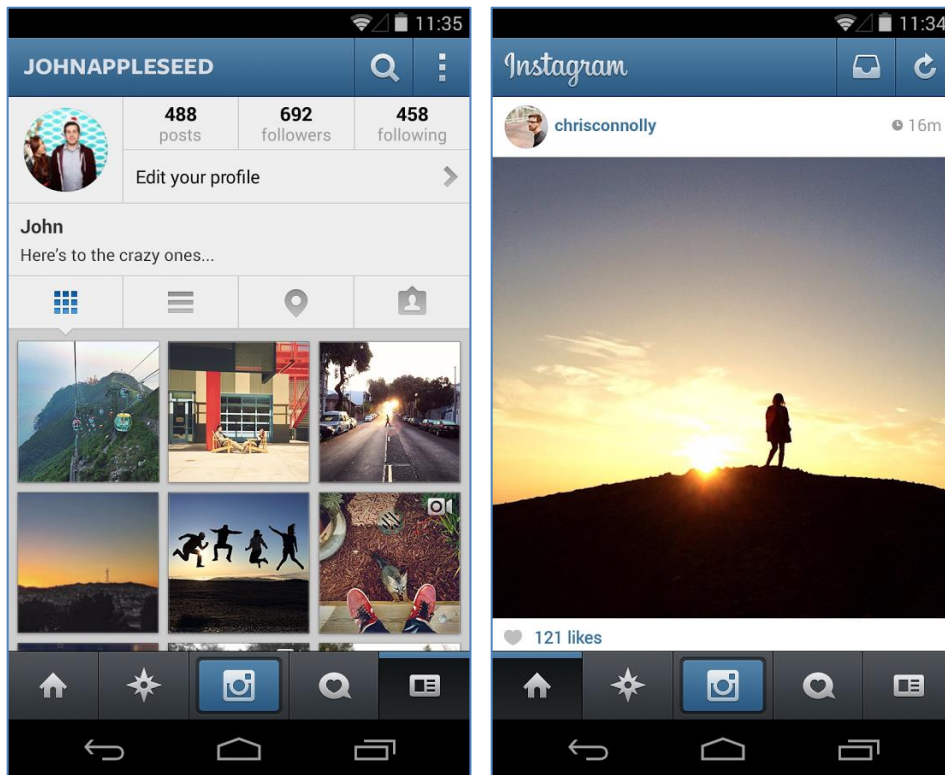


Figure 3 Instagram mobile application (Instagram, Google, 2014)

Instagram is an online mobile social network that allows its users to take pictures and videos, and share them on different networking platforms. Each image will be uploaded with 4:3 aspect ratio and can be changed using several filters (Wikipedia, Instagram, 2014).

Launched in October of 2010, Instagram currently has 20 billion of pictures shared, and every day 60 million new pictures are uploaded and 1.6 billion “likes” are made. In 2011 it was called by Apple “Application of the year”, and in 2012 was acquired by Facebook (Instagram, 2014).

Users of Instagram are approximately divided in Android and iPhone users by 50% and 50%, with 68% female users, conquering 7% of daily photo uploads, among the top four photo-sharing platforms. Instagram is designed for mobile users, having minimalistic interface for uploading pictures comparing to Flickr, which is more suitable for professional photographers (Ferrara, Interdonato and Tagarelli, 2014).

Instagram is also famous for so called “selfies”, a self-portrait photograph, typically taken with a hand-held digital camera or camera phone (Wikipedia, Selfie, 2014). Photos with faces are 38% percent more likely to receive “likes” and 32% more likely to receive comments. At the same time, the number of faces, their age and gender do not affect “likes” (Bakhshi, Shamma and Gilbert, 2014). 46% of photos represent selfies or posing with other friends, 10% can post about food, and approximately the same percent about different gadgets (Hu, Manikonda and Kambhampati, 2014).

Instagram user interface (Figure 3) is similar to Twitter interface. The main interactivity element here is the Like button and each of the photos gets usually more “likes” than actual replies to picture. The author of the picture can be viewed by clicking on his name, and then Instagram will open a list of photos uploaded by the user. The user can also check which pictures are shared around, activity happened to user and whom he/she is following, by clicking buttons in the menu.

### 3.4. Digg

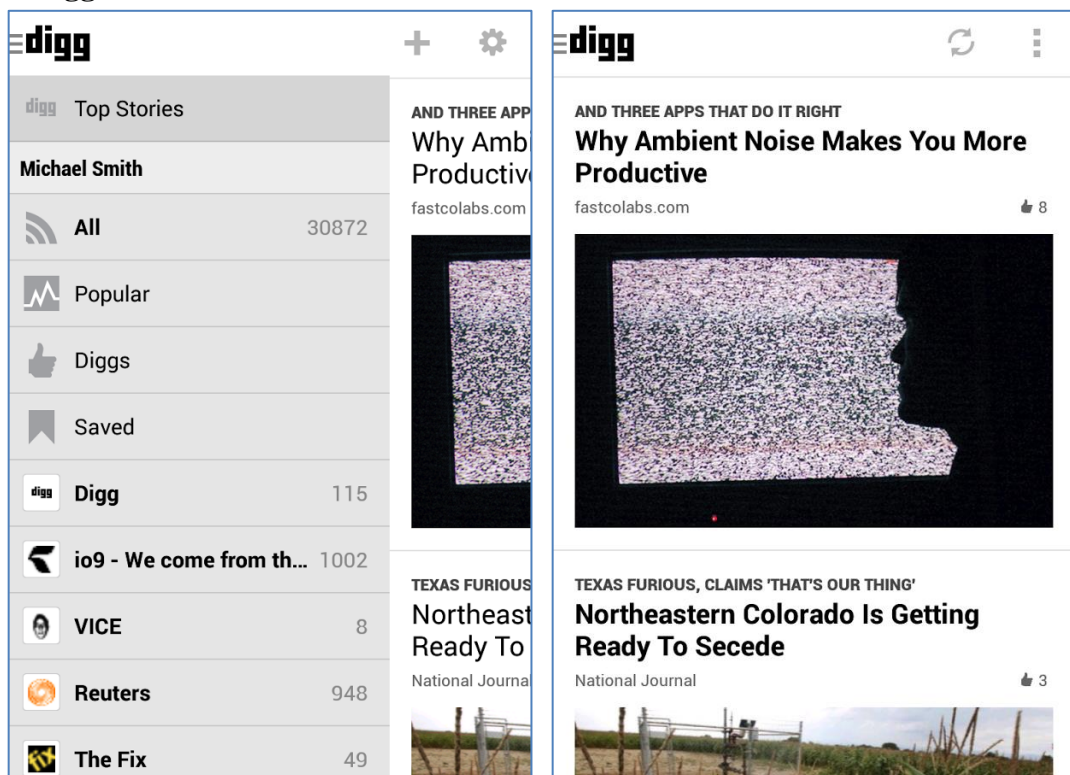
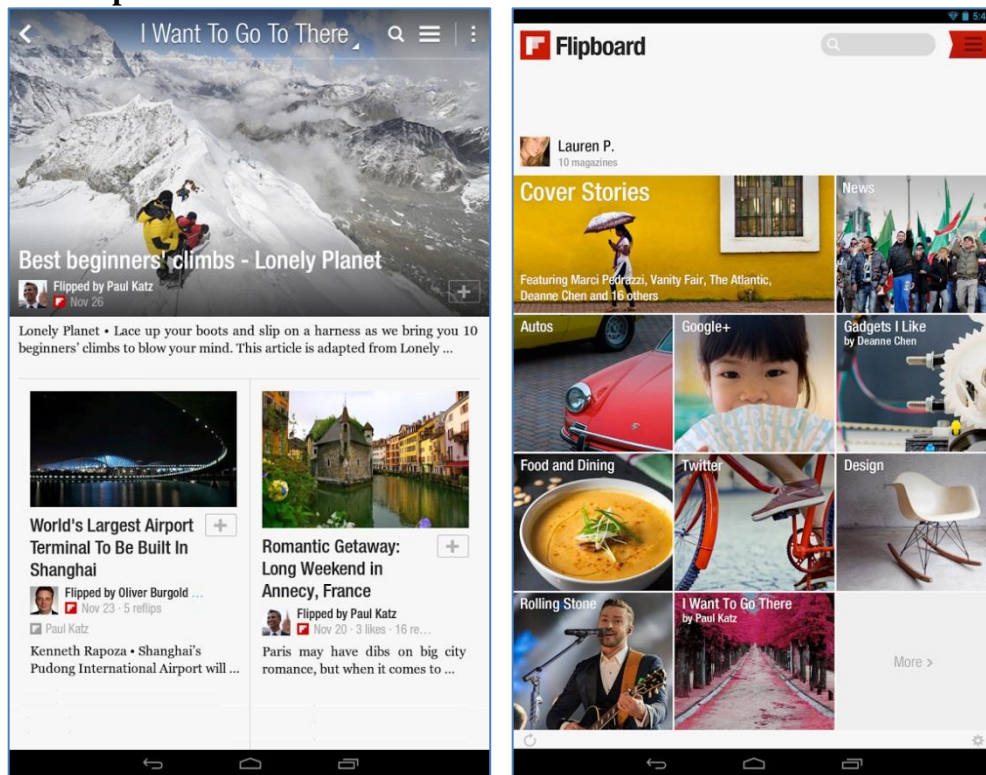


Figure 4. Digg mobile application (Digg, Google, 2014)

*Digg is a news aggregator with an editorially driven front page, aiming to select stories specifically for the Internet audience such as science, trending political issues, and viral Internet issues. It was launched in its current form on July 31, 2012, with support for sharing content to other social platforms such as Twitter and Facebook. (Wikipedia, Digg, 2014).*

The interface (Figure 4) has elements quite similar to other applications and sites, where user can share content: title, preview, category, reference and number of “likes”. It also prompted the creation of other story submission social networks such as reddit (Wikipedia, Digg, 2014).

### 3.5. Flipboard

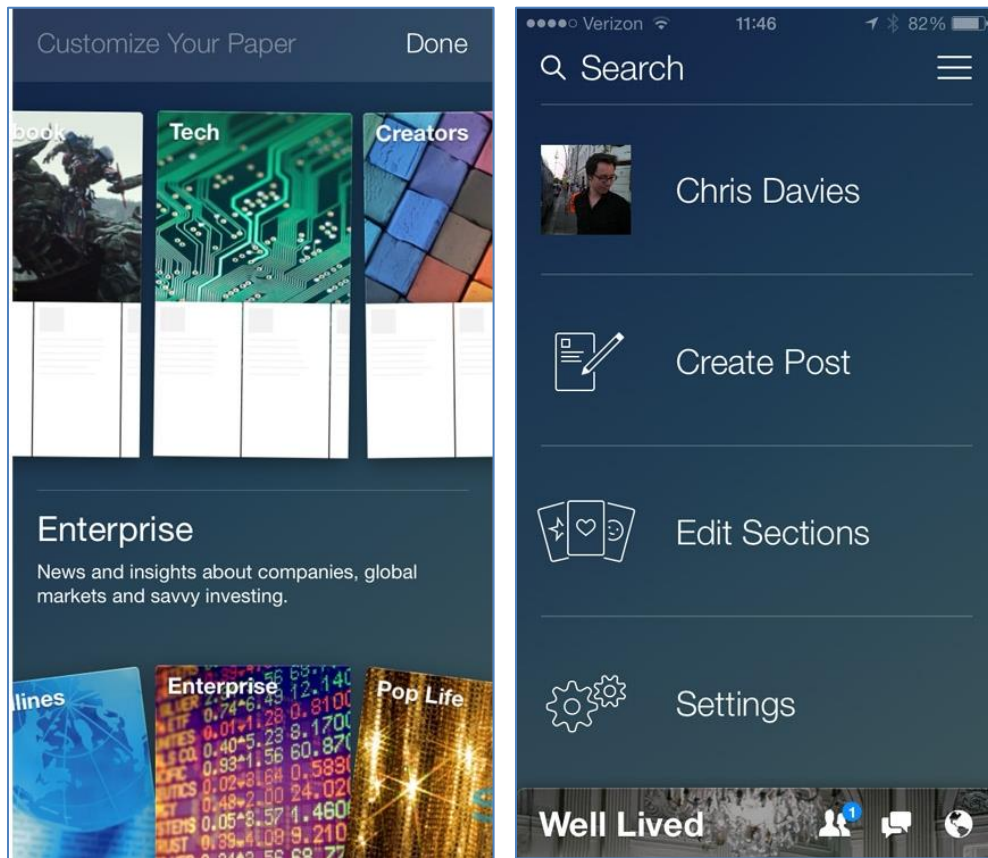


**Figure 5. Flipboard mobile application (Flipboard, Google, 2014)**

*Flipboard is a social-network aggregation, magazine-format mobile app localized in more than 20 languages. The software collects content from social media and other websites, presents it in magazine format, and allows users to "flip" through their social-networking feeds and feeds from websites that have partnered with the company. (Wikipedia, Flipboard, 2014).*

Flipboard combines the idea of print with social media. It supports Facebook and Twitter and lets readers flip through latest stories and updates. The links and images are rendered right in the application, so users do not have to scan long lists of posts, instead they see instantly all stories, comments and images (Flipboard, 2010). The design (Figure 5) consists of previews, titles that are placed on previews and short abstracts of texts.

### 3.6. Paper



**Figure 6. Paper mobile application (Slashgear, 2014)**

Facebook Paper (Constine, 2014) is a standalone mobile app, which delivers full-screen Facebook articles in categories that user likes. The top half of the application shows photos and videos, the bottom half shows status updates and link stories (Figure 6). The user can swipe the screen to move between news. The application has the following categories for reading: Score, Headlines, Cute, Planet, Enterprise, Exposure, Flavor and Ideas. Each category has articles chosen by editors and by the Paper algorithm, meaning that user can see not only content from his timeline, but also editors' picks.

### **3.7. Prototype application**

The aforementioned projects represent rich media in their interfaces in similar ways. For example, a size of media is increasing and gets more attractive. The features such as “like”, favorite and sharing are also common for these applications. This naturally leads to the discussion of how alternative Twitter interface could look like and how the Twitter card feature could be used to expand Twitter features. These questions were discussed in preliminary small scale group interview.

## **4. Preliminary small scale group interview**

### **4.1. Method**

The method of testing the application with two participants at a time was used for initial discussion and understanding of a feature product (Rubin, 2008, p.306). It represents the communication between two people jointly working with the product by understanding how they attack potential problems. By using this method it is possible to capture the conceptual model and thought process of the users.

The method of testing of two participants is an exploratory test that is conducted early in the development cycle in the preliminary stages of being defined and designed. The user and the usage of product were defined, having functional specifications and early prototypes. The objective of the method is to evaluate the effectiveness of this prototype and to understand the user’s conceptual model. It is possible to understand if the user understands the fundamental elements of the interface, how well it represents objects, manipulations and navigation. It is also possible to answer more abstract questions, such as what do users think and conceive about project (Rubin, 1994).

The prototype in this case should not represent the entire functionality of the product, but should be enough for the test goal. The testing process itself is quite informal and collaborative between the participants and test monitor. The users share their ideas how to improve the interface; the emphasis is on discussion and examination of high-level concepts and thought process.

Another reason to choose the method of testing of two participants was also support of user-centered design, “... the practice of designing products so that users can perform required use, operation, service, and supportive tasks with a minimum of stress and maximum efficiency” (Woodson, et al., 1981). It allows to derive product goals, objectives, context, and environment from the user’s viewpoint (Rubin, 2008, p.17).

### **4.2. Implementation**

The group consisted of two active Twitter users (2 males, average age 23), who are using it every day for reading, writing and replying to messages. The session was held in a free discussion format, where some questions were asked and a few



wireframes were shown and responders had to comment freely. The participants were encouraged to communicate with each other during discussion.

The official Youtube application has had an evolution of its design, moving from the larger amount of information about videos and smaller size of video player, to the opposite situation with just a title and big preview image. That is why first question was about it and what happened with design during this time and why do they think did it happen.

The new design was described as “flat”, “trendy”, “more simple with more images”, “images are bigger”, “less different data”. The change of color scheme from black to white was also mentioned, as well as the fact that this kind of design looks more “interactive”. According to the participants, the description of video is not needed that much, since often it doesn’t say anything and the title is more explanatory. Another factor that can be important, when the user is searching for video, is the number of views.

The fact that Youtube and other services are moving to bigger thumbnails and interactive content led to next question – what if the description and title were eliminated and rich media shown directly in a full screen. In other words, what will happen if people’s choice is based on only preview with no textual information. The wireframe with four possible rich contents was shown to represent this situation: video, image, web site preview and audio. The participants mentioned that video and picture have different settings - the preview of video usually requires first description or title to read, in order to understand what this preview is about. The opposite happens for the picture – the user want to see what this picture is about and who uploaded it – after he/she already have seen it. Preview of audio player and websites seems to be self-sufficient to be shown in a full-screen.

Next, the responders got a wireframe representing the tweet elements and numbers, assigned to them for easier discussion. They were asked to say what is the most important in a tweet that affects them to focus their attention on it and to read and interact with this rich media. The highest interest is caused by mentioning the user in the timeline’s thread, meaning that tweet was intended for him. Another important fact that attracts attention is who the author of the message is and how his avatar looks like (which reduces need in reading of name, if the avatar is already known).

This reply led to another discussion – what if all of the Twitter rich media content is shown in a full-screen directly with an opportunity to swipe between them in a mobile phone. The response was negative, mentioning for example that just the rich content itself open in a full screen is not tempting enough to check it. It also raises the problem of order, since chronological order is not relevant anymore here without dates, description and context. One rich content element per page seems to be also too demanding for the attention of responders, since it can be uninteresting for user, but full

screen calls an association with something important and needed. In addition it causes the user to scroll intensively to find something very interesting.

As the previous assumption that full screen can be interesting for users was criticized, the responders were asked to suggest an alternative idea for usage of rich media content and how to choose the tweet that can be interesting for reader. The participants advised to base the selection on messages from your friends, number of retweets, specific hashtags and messages from organized Twitter lists, or from Discovery section, which has recommendations what to read for this user. Facebook was given as an example, as it uses some algorithms and chooses the messages that are important and have to be shown.

Then the participants were asked to suggest the user interface for the full screen - how a tweet with rich content could be shown. They said that seeing the part of next tweet's content is very advisable. On the other hand, the idea of using carousel element, which is widely used in mobile and web environment, was discarded by the participants as taking a lot of space and not creating tempting content. The same happened with the idea of using horizontal scrolling, which was rejected as the vertical scroll was found to be much better and fitting more content.

The idea of using Twitter rich media content in the same way as Instagram appeared during discussion. This means that Instagram could be extended to usage of any type of rich media. It was mentioned also that the new user interface should certainly include the initial description from the tweet which is more important than description from video itself.

The preliminary discussion with the participants revealed many interesting facts about the Twitter application and users' perceptions about it. The results of this discussion enforced to refuse from the idea of showing only tweets with links, show full screen previews on load, and moreover led to the intention to test the current Twitter card feature and to the question how can it be improved.

The prototype application proposed in this study will extend the official Twitter application features by extending a number of supported rich media services in links. The user interface will look similar to the default application and will not confuse users.

## **5. Application development**

### **5.1. Introduction to the system**

A mobile phone was chosen as the platform for developing a prototype. Mobile Twitter users appear to be more engaged with content compared to desktop users. They check Twitter more often than average Twitter users, are 44% more likely to click on links, 66% more likely to retweet and 76% more likely to favorite a Tweet (Twitter Advertising, 2013).

The development of the application was divided in two parts – user interface and interaction logic implementation. Both parts were created using web technologies. The user interface is based on HTML5 and CSS, and the interaction logic on JavaScript. It is possible using PhoneGap (Wikipedia, PhoneGap, 2014), an open source framework that allows creating mobile apps using a standardized web application programming interface (API) for the different platforms, mainly for smartphones.

In this application PhoneGap was a middle layer between JavaScript and the specific API of the phone. It runs a web browser with additional features inside of the application window. The produced code is cross-platform, thus it can be compiled for different operating systems if needed. For testing purposes code was compiled for Android and tested in OnePlus One phone, which was running a built-in operating system Android Runtime Jelly (4.2.2).

An access to the user's timeline and related features was provided by Twitter API, which gives access to a big variety of stored data. For example, the developer can make a search, show the user's timeline, and reply to a tweet, post new tweets and other features. Access to these queries is allowed through the widely used REST API, which can be defined (Wikipedia, Representational state transfer, 2014) as an architectural style consisting of a coordinated set of architectural constraints applied to components, connectors, and data elements, within a distributed hypermedia system.

The application also required the way of creating preview for links in tweets. The solution suggested by Embedly API (2014) was used in order to speed up the development of prototype enough for user testing. Embedly provides HTML, which is ready to be included in the web page, after the developer has provided a link to the site. It supports more than 250 providers of rich content, for example, for video Vimeo, Hulu, for images Instagram, Dribbble, Imgur and for other rich media Google Maps, Storify, Tumblr, Kickstarter, Formspring and many other providers. Embedly is used, for example, by The Guardian, The New York Times, Reddit, AOL and other big media companies. Embedly provides interactive previews or at least a thumbnail for any web page in the internet.

The design was implemented using Bootstrap framework (Bootstrap, 2014), a free collection of tools for creating websites and web applications. It supports responsive web design and contains HTML and CSS-based design templates for typography, forms, buttons, and navigation and other interface components, as well as optional JavaScript extensions.

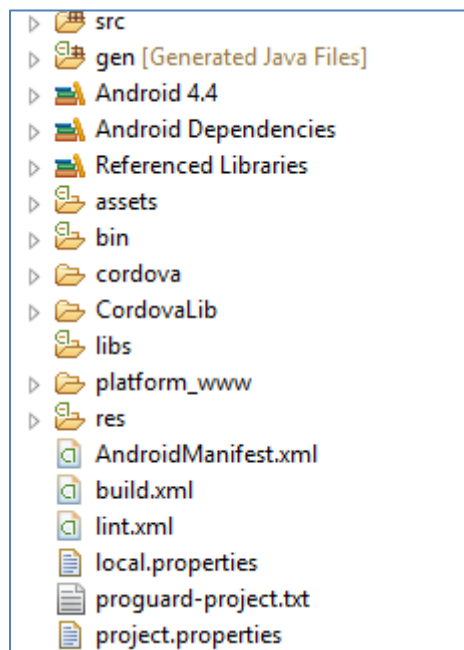
## **5.2. PhoneGap development**

The PhoneGap development for Android can be done with standard integrated development environment (IDE) called Android Developer Tools (ADT). This IDE is

based on open-source Eclipse software tool, which supports a variety of different programming languages. Eclipse is an open development platform for frameworks, tools and runtimes for building, deploying and managing software across the lifecycle.

Each PhoneGap project has from the start predefined files (Figure 7), which allow developer to start an empty application. Later he/she can extend this empty application with own files. The basic structure of PhoneGap Android project can be represented with following files and folders (krasnoff, Code Project, 2012):

- src: Java sources of PhoneGap related files
- gen: Generated Java Files, Android SDK internal use only
- assets/www: All the HTML and JavaScript logic goes here
- AndroidManifest.xml: Android definition file. It has Android minimum version, description of Android device capabilities, to which project has to have access



**Figure 7. Structure of the Android application**

Android Developer Tools give an opportunity to create virtual Android Devices as virtual machines and test the application in an environment close to real mobile phone.

ADT tool is suitable for the standard Android project, but PhoneGap is based on HTML and JavaScript, that is why Aptana plug-in for Eclipse (Aptana, 2014) should be installed in addition. It supports PHP, HTML, CSS and JavaScript development, code assistance, debugging and code highlighting.

### 5.3. Twitter API connection

The connection to Twitter can be established through special Application Programming Interface (API) and it allows developers to use in their application all

corpus of data available in Twitter. It can provide a simple Follow button for the user's web site, or access to complicated queries such as search, showing user's timeline, replying to tweets, posting new tweets and so on.

Access to these queries can be done through the so called REST API, Representational state transfer, which is an alternative to other distributed-computing specifications such as SOAP (Wikipedia, Representational state transfer, 2014).

For the development of the application we need an access to the following API methods (Twitter, 2012):

**GET statuses/home\_timeline**

“Returns a collection of the most recent Tweets and retweets posted by the authenticating user and the users they follow. The home timeline is central to how most users interact with the Twitter service”.

**GET favorites/list**

“Returns the 20 most recent Tweets favorited by the authenticating or specified user”.

**POST favorites/create**

“Favorites the status specified in the ID parameter as the authenticating user. Returns the favorite status when successful”.

**POST favorites/destroy**

“Unfavorites the status specified in the ID parameter as the authenticating user. Returns the unfavorited status in the requested format when successful”.

#### **5.4. Twitter authorization and OAuth**

The Twitter application should support OAuth method of authorization in order to get access to its API (Twitter, OAuth | Twitter Developers, 2014). The OAuth authorization framework appeared (Internet Engineering Task Force, 2012) as an alternative for traditional client-server authentication model, which required the third-party application to store the user's credentials. In addition, it allowed access to user's protected resources with no restrictions and resource owner could not revoke access for this third-party application.

*The OAuth 2.0 authorization framework enables a third-party application to obtain limited access to an HTTP service, either on behalf of a resource owner by orchestrating an approval interaction between the resource owner and the HTTP service, or by allowing the third-party application to obtain access on its own behalf.* (Internet Engineering Task Force, 2012)

The OAuth enabled application gets an access token - a string storing a specific scope, lifetime, and other access attributes (Internet Engineering Task Force, 2012). This token is issued by an authorization server with the approval of the resource owner.

The client uses the access token to access the protected resources hosted by the resource server.

The usual OAuth authorization behavior for website consists of few steps. First, the website redirects user to the special authorization page of API owner. Secondly, the user puts his credentials to API owner form and after successful authorization he/she will be redirected back to initial site.

However if it is not a website but an application, it does not have URL to be redirected to. Twitter provides alternative way for this case – the user receives special pin-code after authorization, then he/she returns manually back to the application and puts in this code.

### **5.5. Data storage**

The application has to keep user's data, when he/she accesses it, for example in order not to ask user again for a password after he/she closed this application. In addition, Twitter OAuth tokens, which are used every time the application makes a request to Twitter server, should be saved. HTML5 already has for it built-in storing tools called web storage, which is supported by PhoneGap, so no additional libraries needed.

*Web storage and DOM storage (document object model) are web application software methods and protocols used for storing data in a web browser. Web storage supports persistent data storage, similar to cookies but with a greatly enhanced capacity and no information stored in the HTTP request header. There are two main web storage types: local storage and session storage, behaving similarly to persistent cookies and session cookies respectively (Wikipedia, Web storage, 2014).*

The lifetime of session storage is restricted by the lifetime of the browser window or tab, where the site is located. This means that it will be cleaned after the browser is closed, which is not suitable for needs of the application that is being described here. That is why local storage should be used instead to save data between sessions.

### **5.6. Embedding media**

Since the prototype application should be able to render previews for links, a third-party tool Embedly was chosen. It supports big amount of media providers and can represent a preview basically for any address in internet as piece of HTML code that can be later embedded to any web page. Embedly represents an oEmbed open format that allows inserting content from one website into another (Wikipedia, oEmbed, 2014)., which is supported by such web sites as Flickr, Hulu, Rdio, SoundCloud, SlideShare and the like.

Embedly consists of 3 services: Embed, Extract and Display, from which we are interested only in Embed, which is actually representing rich media based on its URL.

Embedly has paid and free versions, but for testing purposes the latter is enough, since it allows checking 5000 URLs per month and has just a rate-limit of 15 requests per second.

Embedly supports more than 250 providers of rich content, for example, for video Vimeo, Hulu, for images Instagram, Dribbble, Imgur and for other rich media Google Maps, Storify, Tumblr, Kickstarter and Formspring. What is also noteworthy is that it can basically create a preview of any page, for example for news to show thumbnail and the abstract of the text from this page.

The initial setup consists of registration at Embedly site and getting special key, setting up connection to the API using this key, and handling API response, which contains ready to use HTML code, preview thumbnail and description. In case of sites such as Youtube, SoundCloud and so on, service will return also preview of this content, for example Youtube video player, SoundCloud audio player and user can listen to them directly from the application.

Web sites and applications can connect to API using library for jQuery, which eases up the development process (Embedly, 2014). The basic setup consists of adding jQuery and Embedly jQuery to HTML and it is presented in the Appendix 5.

## **5.7. HTML and CSS libraries**

### **5.7.1. Backbone.js**

Backbone.js (Backbone.js, 2015) is a JavaScript library, which allows creating a structure for a single-page application. It divides the application into few parts and makes a communication between them. The prototype application uses it to follow a certain structure of files so that it will be easier to understand the code. In addition, Backbone.js makes the development faster, since it includes functions frequently used for JavaScript development.

### **5.7.2. Twitter Bootstrap**

One of the ways to speed up application development is using a framework for design of the application. There are a few libraries that provide basic front-end structure for web applications based on CSS and JavaScript. One of the most popular frameworks for this is Bootstrap (licensed under MIT) and it was used for the prototype development.

Applying Bootstrap styles to the web application can be done by inserting links to its CSS and JavaScript, which can be located locally or for example using special content delivery network (Bootstrap, Getting started, 2014). Bootstrap by default supports responsive design, which is useful for the PhoneGap application, because then design can adapt to any Android device screen.

### **5.7.3. jQuery**

Another library that speeds up the development is jQuery. The work with web application means extensive interaction with elements of HTML in JavaScript. jQuery is widely known among developers and it is used for the development of the prototype in this study. It allows navigating easily through the HTML document structure with special selectors, working with events and AJAX requests. Moreover, some of the plugins mentioned here are extensions for jQuery, so it is actually needed by plugins used in the prototype application.

### **5.7.4. twig.js**

The application code grows significantly during the development of the project and it can become unintelligible, if a code structure was not planned beforehand. One of the ways to avoid it is to move frequently repeated parts of the HTML markup to special templates.

twig.js (Roepke, 2014) is JavaScript implementation of Twig templating language made for PHP, mainly used in a very popular Symfony2 PHP framework. The usage of the template system reduces the amount of code and makes it more readable, since repeatable parts of code are moved out to separate files. Some simple logic inside of the template is also possible, such as comparison operators, loops “for”, variable setting and reading.

## **5.8. AJAX loading indicator**

The process of getting a preview for websites consists of two steps: a request to Embedly API and request to elements of preview itself. Taking this into account, it is a good idea to have some indicator of how the process is going so that user will understand in which stage he/she is.

The SVG indicator “Loading...” (Jackson, 2014) was used for these purposes, since it is licensed under MIT and SVG format supported by modern browsers in vector format and thus responsive to changes of height and width without quality loss.

## **5.9. Twitter API libraries**

### **5.9.1. Codebird.js**

The prototype application needs to connect to Twitter API and thus, it requires a library for it. Codebird.js (Jublo Solutions, 2014) is a Twitter library written in JavaScript, which allows calling API requests to Twitter directly from the browser and has built-in support for OAuth user authentication. First, the prototype application should authenticate API requests on behalf of a certain Twitter user, according to OAuth instructions. This is done by creating a pin input field, opening a new window in



browser that will ask for user credentials at site of Twitter. The user gets pin-code, returns to the application and puts there the received pin-code.

### **5.9.2. jStorage**

Browsers' default storage supports just strings. On the other hand, the prototype application should have an ability to save the data, for example, authentication data and cache. The objects returned by Twitter are usually objects or arrays of objects and thus the application needs some alternative solution that could work with it.

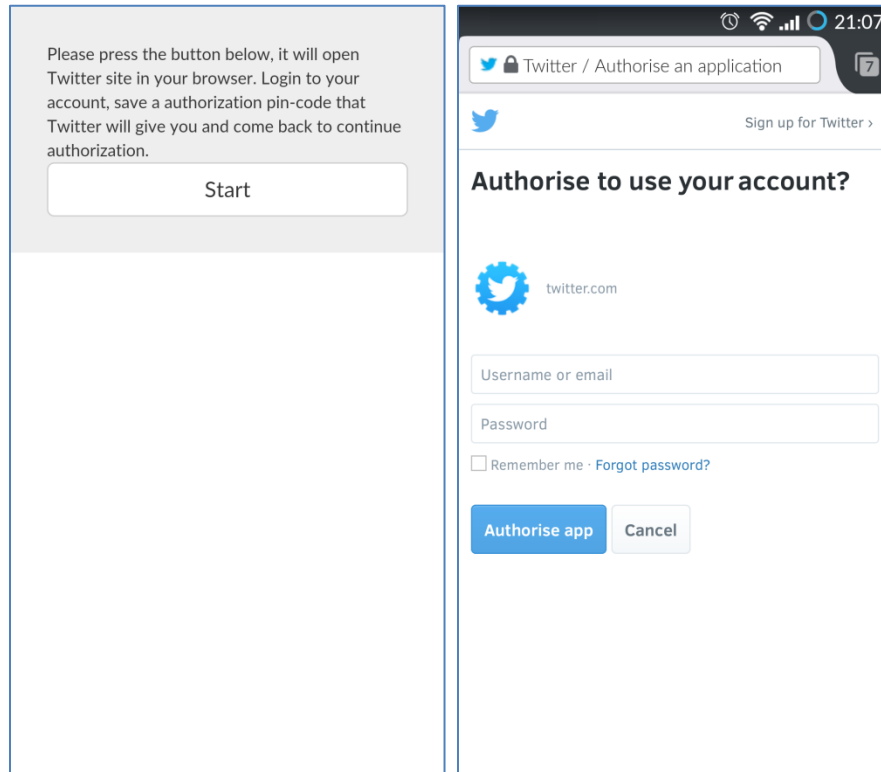
jStorage (Reinman, 2012) is a cross-browser key-value database to store data locally in the browser. jStorage supports all major browsers, both in desktop and in mobile. Additionally, jStorage is library agnostic, it works well with any other JavaScript library on the same webpage, if it is jQuery, Prototype, MooTools or any other framework. jStorage makes use of HTML5 local storage where available and userData behavior in Internet Explorer older versions. jStorage can store strings, numbers and JavaScript objects. The latter one, for example, is not supported by default in browsers; it has to be transformed first to the string using JSON encoding.

### **5.9.3. twitter-text.js**

Any Twitter client has to be able to parse data that is contained in the tweet, for example a text link that should become a real "clickable" hypertext link. A hashtag should be transformed to a search interface link. Twitter-text utility (Twitter, Twitter Text Libraries, 2014) provides such text processing routines for tweets.

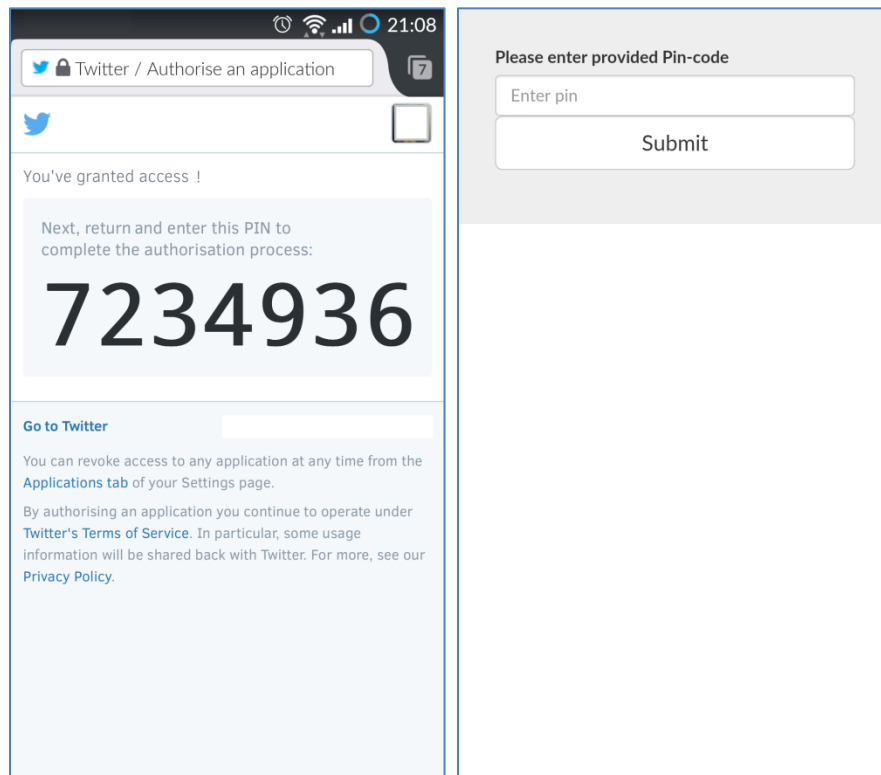
## 6. Prototype user interface

### 6.1. Authorization process



**Figure 9. First and second steps of authorization**

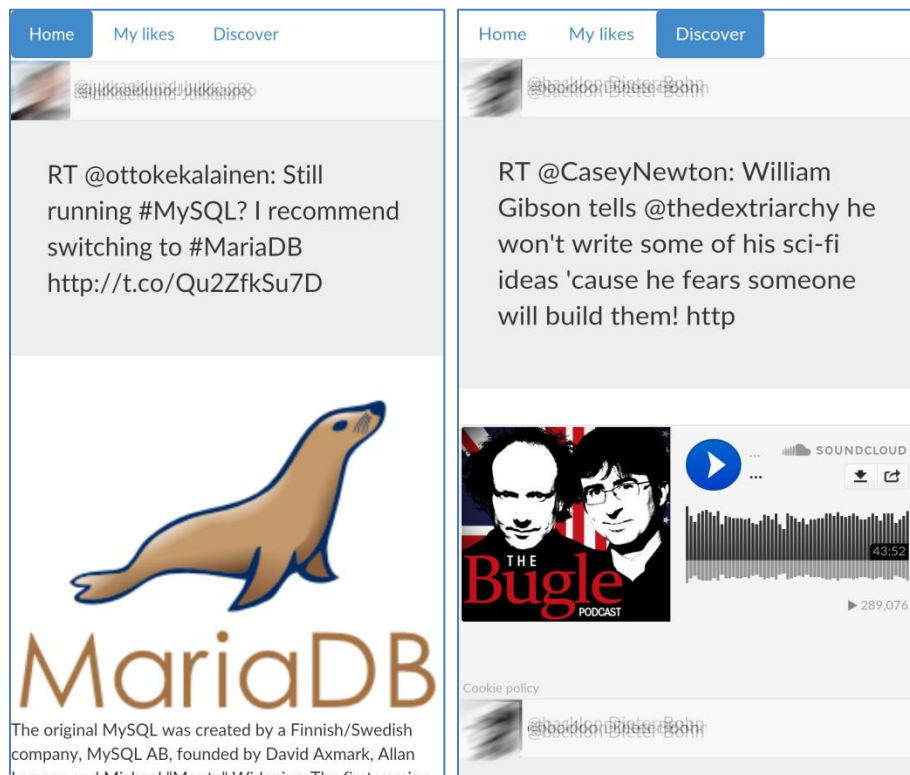
The authorization consists of 4 steps (Figures 9 and 10). First, when the user opens the application he sees the welcome screen, where the description text and then button to start the authorization are located. The application sends a request to Twitter and generates a link after the user clicked the button. The default browser of the phone is then opened asking to input user name and password to continue the process.



**Figure 10. Third and fourth steps of authorization**

Next the user receives a special pin-code, which he/she should input to the application and then return to the application using standard methods of switching between applications in the Android operating system. After the user is done, he will see the main interface of the application.

## 6.2. Main interface



**Figure 8. The prototype application screenshots**

The result application consists of three navigational tabs (Figure 8): Home, My likes and Discover. Home is the user's timeline and looks similar to the official Twitter application. Each of the tweets has an image of the author, his nickname and real name, the text of the tweet and preview of the link (if the text contains any). The Home timeline is scrollable with a swipe of a finger from bottom to the top of the screen (and vice versa to check the latest tweets). My likes feature loads all the favorited tweets that were created by user. The Discover tab is fictional in this case and is the same for all of the users (and thus is not the same as in the Twitter official application). It shows some of the most popular rich media services: SoundCloud, YouTube, Instagram and Flickr, and how they could look like if they could appear in user's timeline. It was important to create this fictional tab, since during user testing it is not feasible to find rich media services from the real user's timeline in the concrete time when the testing takes place.

The application had also an extended version, which had the Favorite button and a number of favorites by other users, in addition to text of the tweet. It was represented in both Home and My likes views. The button had an image of a heart without any text. In addition, the timeline was filtered and showed only tweets that had links. The

extended version of the application was used for separate evaluation and described in the next chapter.

## **7. User testing**

### **7.1. Methods**

The method consisted of two main parts: first, “think aloud” method was used to discuss the Twitter card feature while testing the prototype application and the experience that users have had during testing and earlier in everyday life. Second, the participants filled an experience questionnaire AttrakDiff™ to test how they perceive the prototype application and the idea of increasing number of supported services by Twitter card. AttrakDiff™ is a 23-word Likert scale with opposite pairs in each end of the scale, representing attractiveness and visual appeal qualities, created to evaluate the usability and design of interactive product. In the end of the experiment, the moderator can check the results with an automatically generated graphic and text related to the research. A comparison method was used in the AttrakDiff™ questionnaire, which relates two versions of the product before and after changes.

The participants were divided into three groups: pilot testing with one of the participants to reveal possible issues that can appear during testing, the core testing of the application prototype with five participants and the testing of the extended version of the application with additional features and one participant.

The first participant contributed in pilot testing, where all the technical questions related to the future user testing were verified: if the participant understand the terms in the questionnaires, how to start the recording application in the phone, how to remove all the private data of the user in end of the session, how long these sessions actually take and so on. The pilot testing results are not detailed, since it was held just for preparations for actual testing with other participants and for solving common mistakes. This participant passed through the same steps as in real user testing: giving instructions, replying to questionnaires and discussion of the application.

The next five participants were the core of this research; they answered the questions about their Twitter timeline, tested the application prototype and compared it with standard Twitter client by filling the user experience questionnaires.

The last participant tested the extended version of the application, which had the same features as original prototype with some additions. For example, all his timeline was filtered and showing only tweets with links. The participant was suggested to imagine that presented timeline is a separate application for sharing links with friends and liking them. Like button had in addition the number of times that it was favorited by other users.

All in all, user testing of the prototype was held among 6 workers of web agencies (6 males, found among friends and co-workers, average age 29) located in Helsinki in Finland, leading in open source web design and development. All interviewed participants were active Twitter users, and working in development and project management. Predictably, all of them had excellent computer skills, using computer and Internet daily for different purposes (playing music, using social networks, playing games, sending and receiving e-mails and so on).

## **7.2. Results of think aloud method**

The think aloud method in this study extended the user experience questionnaire, since it did not contain any comments from the participants and it could be difficult to interpret the results. They were asked, for example, to tell if the preview engages the participants to interact with the tweets, how they use the Favorite feature and whether the preview affects the usage of this feature. In addition, they told their opinion about the tweet with and without preview, and about the representation of the preview in the official and prototype applications. The background questionnaire, consent form and test plan with questions are provided in the Appendixes 1, 2 and 3. The short transcription of the think aloud sessions is provided in Appendix 4. The main ideas that were expressed during think aloud method are mentioned here.

First, two participants suggested the idea of giving an opportunity to hide expanded tweets for those people who do not like it. The concerns were connected with unnecessary interactions that can be caused by it – animated images that can attract too much attention, occasional clicks on videos while browsing timeline. It is worth to mention that the current prototype outputs all the previews directly to the timeline. The official application shows just part of preview and the user has to click on it to see the full preview.

One of the participants appreciates in Twitter only textual data and it seems that improvements in Twitter card were not affecting him much, since this feature itself is not used at all.

Three participants mentioned that the feature of showing short abstract for the link was really interesting for them. This is a built-in feature of Embed.ly and was not initially considered as a subject for this research and thus it became a surprising discovery.

Some of the services were wrongly scaled to the phone screen for unknown reason. For example, some of YouTube videos looked too small and almost impossible to interact with. This should be explored in future versions of this application.

During the research the participants' timelines were usually filled by thumbnails of links, quite rarely there were any really interactive links such as YouTube or Flickr. However, when they checked examples of how these services could look like in

timeline, they seem to appreciate this more than thumbnails. They mentioned that these previews can save time, since the user does not have to click on the link and wait while Twitter opens a site in a separate window of the browser application.

Quite common was the opinion that it should be some balance between the number of links that are supported by Twitter card and safety. Uncontrolled number of links that can be shown for user can terrify him because of its uncertainty. On the other hand, Twitter supports just few popular services and the situation when Instagram was removed from supported services for unknown reason the participants found disadvantageous for the end user.

There was no clear picture of how the participants are using favorites – it can be either as bookmarks, or as “like”. Three participants mentioned that they can bookmark the tweet even before reading the link itself, just by looking at preview in Twitter card. It can happen for example if they want to check link later, but now they are in the bus travelling somewhere and cannot read properly the full text. On the other hand, those who favorite as in a “like” feature have many different motivations to do this. Twitter card can make this action more appealing, but still it is just one of many reasons for doing so. Because of this ambiguity it is hard to assume any correlation between improvements in Twitter card and behavior in favoriting.

### **7.3. Results of questionnaire**

AttrakDiff™ is an instrument for measuring the attractiveness of interactive products. With the help of pairs of opposite adjectives, users (or potential users) can indicate their perception of the product. These adjective-pairs make a collation of the evaluation dimensions possible. The following product dimensions are evaluated (AttrakDiff™, 2014):

**Pragmatic Quality (PQ):**

“Describes the usability of a product and indicates how successfully users are in achieving their goals using the product”.

**Hedonic quality - Stimulation (HQ-S):**

“Mankind has an inherent need to develop and move forward. This dimension indicates to what extent the product can support those needs in terms of novel, interesting, and stimulating functions, contents, and interaction- and presentation-styles”.

**Hedonic Quality - Identity (HQ-I):**

“Indicates to what extent the product allows the user to identify with it”.

**Attractiveness (ATT):**

“Describes a global value of the product based on the quality perception. Hedonic and pragmatic qualities are independent of one another, and contribute equally to the rating of attractiveness”.

Figure 9 represents the results that were automatically generated by AttrakDiff™, bottom and left values are lower. The “V” rectangle represents the Twitter card feature in default Twitter client. The “N” rectangle is the prototype version of Twitter card. This figure clearly shows that participants’ experiences towards the default application are quite neutral and not engaging. In addition, the prototype application clearly moved the perception towards hedonic and pragmatic qualities. At the same time the square of the rectangle reduced, which means that the users are less variable in the evaluation of new Twitter card opportunities.



**Figure 9. Portfolio with average values of the dimensions PQ and HQ and the respective confidence rectangles of the product in the fore-study and after-study**

The picture of change in these values is represented in Figure 10 as a line graph. The best results were achieved in “Hedonic quality – Stimulation” and “Hedonic Quality - Identity”. It is worth to mention here that the prototype improved statistically significantly only in the stimulation aspect of hedonic quality. Other values did not succeed in it, however they improved in absolute values. In addition, the small number of the participants could affect these results.



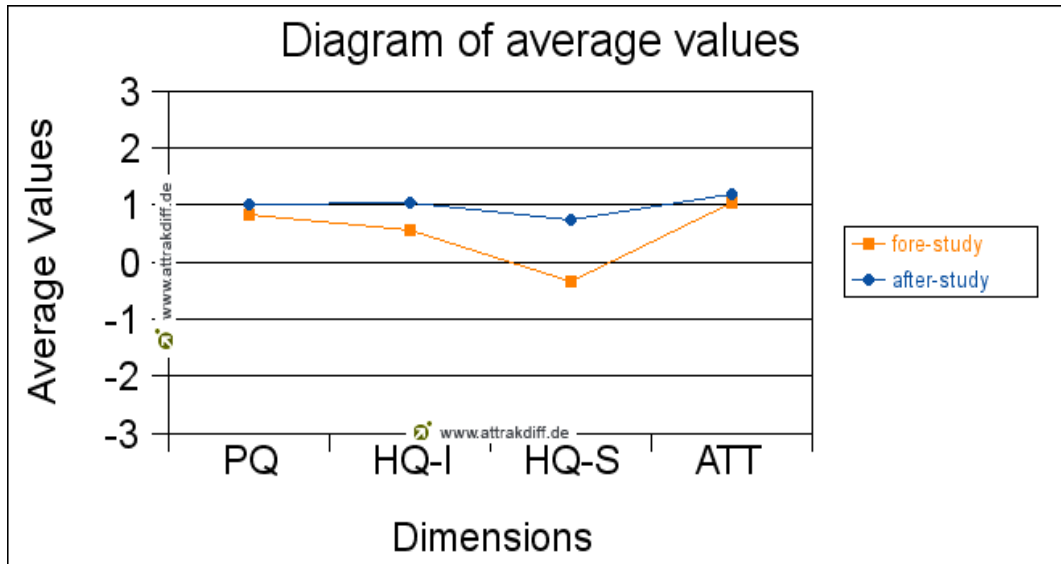


Figure 10. Mean values of the four AttrakDiff™ dimensions

The exact mean values of the word pairs are presented in Figure 11. Among all values the “undemanding - challenging” word pair seems to be the extreme value. It is hard to say if such high value for the prototype application in being undemanding is advantage or the opposite.

It is interesting that the values in the Attractiveness section stayed almost unchanged. This means that the prototype application did not have any significant effect in this field. It can be explained by the fact that design of Twitter application was created by professionals and they put an effort to make this application attractive. On the other hand, the Pragmatic quality values were changed but in different directions among the X-axis. However, the average value stayed unchanged, which can be considered as good result as well. The prototype application was described as “inventive”, “human”, “connective”, “integrative”, “novel”, “captivating” and “pleasant”, but also “complicated”, “unpredictable” and “repelling” and needs an improvement in these areas.

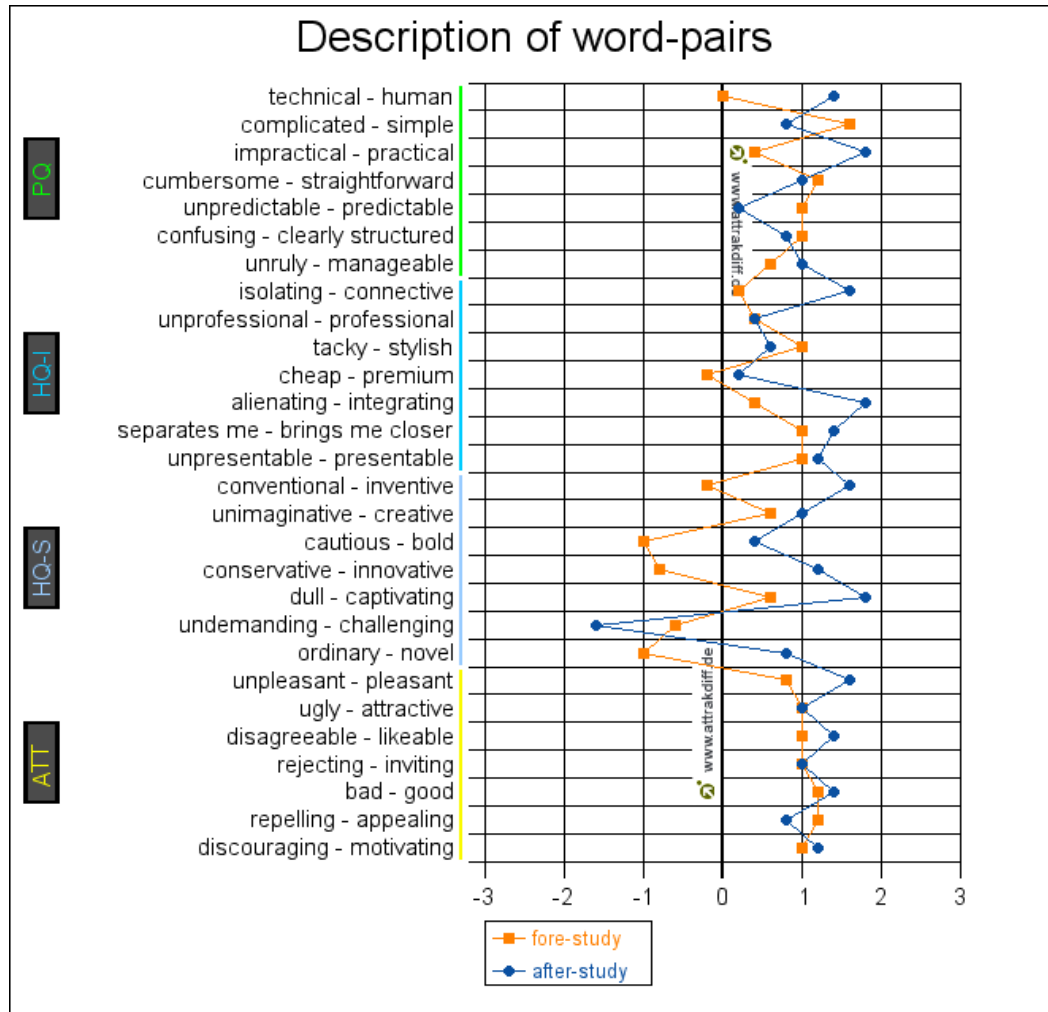


Figure 11. Mean values of the AttrakDiff™ word pairs

## 8. Discussion

All in all, the user-testing participants liked the idea of the application and even asked if the development would also be continued after it. The results from AttrakDiff™ user experience evaluation tool showed an evident increase in hedonic satisfaction and one of them was statistically significant. However, due to the small number of the participants, it is hard to estimate how true that is. A higher number of participants could improve exactness of results. The participants appreciated the extended number of supported services in Twitter card.

During the user testing one characteristic property of Twitter was found. It was revealed that the usage of Favorite button varies among users – some use it as a bookmark, and some as “Like”. There was no clear preference over it, but this could be explored in further research. Partly because of this ambiguity there was found no effect of the Favorite button on user’s interaction with Twitter card.

The application was lacking the real Discover functionality, which could show the data that user sees in his Twitter timeline. The participants expressed interest in seeing

real data there but this feature is not available currently for developers through Twitter API.

The results of user testing can be used for making improvements in the current user interface of Twitter, since the participants appreciated the extended number of supported services in Twitter card. In addition, the source code of the application can be used also for creating a separate Twitter client that will show only tweets with links and their preview, or for a separate social network that shares interesting links with other users through a mobile phone with a broad support of the previews in messages.

AttrakDiff™ results also can be used for exploring Twitter card feature itself and what could be improved or how users perceive this feature. The results of discussions can also be taken into account for feature improvements in Twitter interface, if the developer of the application considers it from the point of view of user-centered design.

Strategies to enhance the prototype application might involve extending its functionality. For example, it could also provide visual information for tweets without links. This can be achieved, for instance, using text categorization and extracting pictures related to this category through image search system. The application could be ported to a desktop as well, providing the same features there. This could open it up to a broader audience and make easier configurations in a computer. Another idea is handling multiple links in a tweet, instead of just taking the first one from it. In future, the application could show a few previews according to the number of links the tweet has. In addition, the prototype application could use its own code that could generate previews, at least for the most popular web services.

The source code (Appendix 5) can be used for future development of program that can work with Twitter and Embedly API, providing extended features as whole or as module in the more complex application.

## References

- Akcora, C.G., Demirbas M., 2010. Twitter: Roots, Influence, Applications. Buffalo, NY, USA: Department of Computing Science and Engineering State University of New York at Buffalo.
- Alexa. 2014 Alexa - twitter.com. [online] Available at: <<http://www.alexa.com/siteinfo/twitter.com>> [Accessed 24 August 2014].
- Aptana. [online] Available at: <<http://www.apтана.com>> [Accessed 24 August 2014].
- AttrakDiff™. [online] Available at: <<http://www.attrakDiff.de/index-en.html>> [Accessed 30 December 2014].
- Backbone.js. [online] Available at: <<http://www.backbonejs.org>> [Accessed 15 March 2015].
- Bakhshi, S., Shamma, D.A., Gilbert, E., 2014. Faces Engage Us: Photos with Faces Attract More Likes and Comments on Instagram. Proceedings of the SIGCHI Conference on Human Factors in Computing System, CHI '14. Toronto, Canada, April 26 - May 1. New York, NY, USA: ACM.
- Bootstrap. Bootstrap - The world's most popular mobile-first and responsive front-end framework. [online] Available at: <<http://www.getbootstrap.com>> [Accessed 30 December 2014].
- Bootstrap. Getting started. [online] Available at: <<http://getbootstrap.com/getting-started/>> [Accessed 30 December 2014].
- Boyd, D., Golder, S., Lotan, G., 2010. Tweet, Tweet, Retweet: Conversational Aspects of Retweeting on Twitter. 43rd Hawaii International Conference on System Sciences (HICSS). Kauai, HI, USA, January 5-8, 2010. Washington, DC, USA: IEEE.
- Carle, S., 2012. More interactive Tweets, in more than 2000 ways. [online] 25 October 2012. Available at: <<https://blog.twitter.com/2012/more-interactive-tweets-in-more-than-2000-ways>> [Accessed 30 December 2014].
- Chen, K., Chen, T., Zheng, G., Jin, O., Yao, E., Yu, Y., 2012. Collaborative Personalized Tweet Recommendation. Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval. Portland, Oregon, USA, August 12–16, 2012. New York, NY, USA: ACM.
- Choudhury, M.D., Gamon, M., Counts, S., Horvitz, E., 2013. Predicting Depression via Social Media. Proceedings of 7<sup>th</sup> International AAAI conference of weblogs and social media. Boston, USA, 8-11 July 2013. [Accessed 15 March 2015].
- Chun, C.W., Lee, J.T., Lee, S.W., Rim, H.C. 2012. Detecting Informative Messages. Based on User History in Twitter. Proceedings of 8th Asia Information Retrieval Societies Conference, AIRS 2012. Tianjin, China, December 17-19, 2012. Berlin, New York: Springer Berlin Heidelberg.

- CodeProject, krasnoff, 2012. Basic structure of an Android project, [online] 31 May 2012. Available at: <<http://www.codeproject.com/Articles/395614/Basic-structure-of-an-Android-project>> [Accessed 28 August 2014].
- Constine, J., 2014. [Online] 30 January 2014. Facebook Announces Paper, A Curated Visual News Reader Launching Feb. 3 On iOS. [online] Available at: <<http://techcrunch.com/2014/01/30/facebook-paper/>> [Accessed 30 December 2014].
- Counts, S., Fisher, K., 2011. Taking It All In? Visual Attention in Microblog Consumption. Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media. Barcelona, Spain, July 17-21, 2011. Menlo Park, California: AAAI Press.
- Crawford, C., 2014. How informative is Twitter, [online] 8 January 2010. Available at: <<http://blog.textwise.com/2010/01/08/how-informative-is-twitter>> [Accessed 30 December 2014].
- Digg, Google, 2015. Digg – Android Apps on Google Play. [image online]. Available at: <<https://play.google.com/store/apps/details?id=com.diggreader&hl=en>> [Accessed 15 March 2015].
- Dix, A., Finlay, J., Abowd, Beale, R., 2004. Human - Computer interaction. Prentice Hall.
- Embedly. Embedly makes your content more engaging and easier to share. [online] Available at: <<https://www.embed.ly>> [Accessed 30 December 2014].
- Ferrara, E., Interdonato, R., Tagarelli, A. Online popularity and topical interests through the lens of Instagram. Santiago, Chile, 1–4 Sep 2014. Proceedings of the 25th ACM conference on Hypertext and social media. New York, NY, USA: ACM.
- Flipboard, 2010. Flipboard Launches World's First Social Magazine. [online] Available at: <<https://about.flipboard.com/newsroom/releases/flipboard-launches-worlds-first-social-magazine/>> [Accessed 30 December 2014].
- Flipboard, Google, 2015. Flipboard – Android Apps on Google Play. [image online] Available at: <<https://play.google.com/store/apps/details?id=flipboard.app&hl=en>> [Accessed 15 March 2015].
- Honeycutt, C., Herring, S., 2009. Beyond Microblogging: Conversation and Collaboration via Twitter. Proceedings of the Forty-Second Hawaii International Conference on System Sciences (HICSS-42). Waikoloa, Big Island, Hawaii, HI, USA, January 5-8, 2009. Los Alamitos, CA: IEEE Press.
- Hu Y., Manikonda L., Kambhampati, S., 2014. What We Instagram: A First Analysis of Instagram Photo Content and User Types. Proceedings of Eighth International AAAI Conference on Weblogs and Social Media. Ann Arbor, Michigan, USA, June 1–4, 2014. Palo Alto, California: AAAI Press.

- Instagram, 2014. Press Page - Instagram. [online] Available at: <<http://instagram.com/press/>> [Accessed 30 December 2014].
- Instagram, Google, 2015. Instagram – Android Apps on Google Play. [image online] Available at: <<https://play.google.com/store/apps/details?id=com.instagram.android&hl=en>> [Accessed 15 March 2015].
- Internet Engineering Task Force (IETF). The OAuth 2.0 Authorization Framework. [online] Available at <<https://tools.ietf.org/html/rfc6749>> [Accessed 30 December 2014].
- Jackson, B., 2014 Loading... [online] Available at: <<https://github.com/jxnblk/loading>> [Accessed 30 December 2014].
- Jansen, B.J., Zhang, M., Sobel, K., Chowdury, A., 2009. Twitter Power: Tweets as Electronic Word of Mouth. *Journal of the American Society for Information Science and Technology*. Volume 6 Issue 11, November 2009, p.2169-2188.
- Jublo solutions. codebird-js. [online] Available at: <<https://github.com/jublonet/codebird-js>> [Accessed 30 December 2014].
- Lake, C., 2014. 18 pivotal web design trends for 2014, [online] 13 January 2014. Available at: <<https://econsultancy.com/blog/64096-18-pivotal-web-design-trends-for-2014>> [Accessed 30 December 2014].
- Lunden, I., 2012. Analyst: Twitter Passed 500M Users in June 2012, 140M Of Them In US; Jakarta ‘Biggest Tweeting’ City, [online] 30 July 2012. Available at <<http://techcrunch.com/2012/07/30/analyst-twitter-passed-500m-users-in-june-2012-140m-of-them-in-us-jakarta-biggest-tweeting-city>> [Accessed 24 August 2014].
- Meier, F., Elsweiler, D., Wilson, M.L., 2014. More than Liking and Bookmarking? Towards Understanding Twitter Favouriting Behaviour. *Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media*. Ann Arbor, Michigan, USA, June 1–4, 2014. Palo alto, California: AAAI Press.
- Naaman, M., Boase, J., Lai, C.H., 2010. Is it Really About Me? Message Content in Social Awareness Streams. *Proceedings of the 2010 ACM conference on Computer supported cooperative work*. Savannah, Georgia, USA, February 6–10, 2010. New York, NY, USA: ACM.
- Nardi, B.A., Schiano, D.J., Gumbrecht M. and Swartz L., 2004. Why we blog? *Communications of the ACM*, Volume 47 Issue 12, pp.41-46.
- Reinman, A., 2012. jStorage - simple JavaScript plugin to store data locally. [online] Available at: <<https://support.twitter.com/entries/166337-the-twitter-glossary>> [Accessed 30 December 2014].
- Roepke, John, 2014. Justjohn/twig.js. [online] Available at: <<https://github.com/justjohn/twig.js>> [Accessed 30 December 2014].

- Rogers, R., 2013. Debanalizing Twitter: The Transformation of an Object of Study. Proceedings of the 5th Annual ACM Web Science Conference. Paris, France, May 2–4, 2013. New York, NY, USA: ACM.
- Rogers, S., 2014. What fuels a Tweet’s engagement? [online] Twitter. Available at: <<https://blog.twitter.com/2014/what-fuels-a-tweets-engagement>> [Accessed 30 December 2014].
- Rubin, J., 1994. Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests. Wiley.
- Schmierbach, M., Oeldorph-Hirsch, A., 2012. A little bird told me, so I didn’t believe it - Twitter, credibility, and issue perceptions. *Communication Quarterly*. Volume 60, Issue 3, 2012.
- Shneiderman, B., Plaisant C., 2005. Designing the User Interface: Strategies for Effective Human-Computer Interaction. Addison-Wesley Publ. Co.
- SlashGear, 2015. Facebook Paper review: Zuckerberg finally “gets” mobile. [image online] Available at: <<http://www.slashgear.com/facebook-paper-review-zuckerberg-finally-gets-mobile-03315551/>> [Accessed 15 March 2015].
- The jQuery Foundation. jQuery. [online] Available at: <<https://www.jquery.com>> [Accessed 30 December 2014].
- Twitter Advertising, 2013. New Compete study: Primary mobile users on Twitter, [online] 11 February 2013. Available at: <<https://blog.twitter.com/2013/new-compete-study-primary-mobile-users-on-twitter>> [Accessed 30 December 2014].
- Twitter, 2012. Documentation: Getting started. [online] 2012. [Accessed 28 08 2014] Available at: <<https://dev.twitter.com/start>> [Accessed 30 December 2014].
- Twitter, 2013. Display Requirements | Twitter Developers. [image online] Available at: <<https://dev.twitter.com/overview/terms/display-requirements>> [Accessed 15 March 2015].
- Twitter, 2014. OAuth | Twitter Developers. [online] Available at: <<https://dev.twitter.com/oauth>> [Accessed 30 December 2014].
- Twitter, Google, 2015. Twitter - Android Apps on Google Play. [image online] Available at: <<https://play.google.com/store/apps/details?id=com.twitter.android&hl=en>> [Accessed 15 March 2015].
- Twitter. About Twitter's link service (<http://t.co>). [online] Available at: <<https://support.twitter.com/entries/109623-about-twitter-s-link-service-http-t-co>> [Accessed 24 August 2014].
- Twitter. Twitter Help Center | The Twitter Glossary. [online] Available at: <<https://support.twitter.com/entries/166337-the-twitter-glossary>> [Accessed 30 December 2014].

- Wikipedia, Bootstrap (front-end framework). [online] Available at: <[https://en.wikipedia.org/wiki/Bootstrap\\_\(front-end\\_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework))>. [Accessed 30 December 2014].
- Wikipedia, Digg. [online] Available at: <<https://en.wikipedia.org/wiki/Digg>> [Accessed 30 December 2014].
- Wikipedia. Flipboard. [online] Available at: <<https://en.wikipedia.org/wiki/Flipboard>> [Accessed 30 December 2014].
- Wikipedia. Information overload. [online] Available at: <[https://en.wikipedia.org/wiki/Information\\_overload](https://en.wikipedia.org/wiki/Information_overload)> [Accessed 30 December 2014].
- Wikipedia. Instagram. [online] Available at: <<https://en.wikipedia.org/wiki/Instagram>> [Accessed 30 December 2014].
- Wikipedia. Microblogging. [online] Available at: <<https://en.wikipedia.org/wiki/Microblogging>> [Accessed 30 December 2014].
- Wikipedia. oEmbed. Available at: <<https://en.wikipedia.org/wiki/OEmbed>> [Accessed 30 December 2014].
- Wikipedia. Representational state transfer. [online] Available at: <[https://en.wikipedia.org/wiki/Representational\\_state\\_transfer](https://en.wikipedia.org/wiki/Representational_state_transfer)> [Accessed 30 December 2014].
- Wikipedia. Selfie. [online] Available at: <<https://en.wikipedia.org/wiki/Selfie>> [Accessed 30 December 2014].
- Wikipedia. Twitter. [online] Available at: <<https://en.wikipedia.org/wiki/Twitter>> [Accessed 30 December 2014].
- Wikipedia. Web storage. [online] Available at: <[https://en.wikipedia.org/wiki/Web\\_storage](https://en.wikipedia.org/wiki/Web_storage)> [Accessed 30 December 2014].
- Woodson, W., Wesley, E. Human Factors Design Handbook: Information and Guidelines for the Design of Systems, Facilities, Equipment, and Products for Human Use. New York, McGraw-Hill, 1981.
- Wu, S., Hofman, J.M., Mason, W.A., Watts, D.J., 2011. Who Says What to Whom on Twitter. Proceedings of the 20th international conference on World Wide Web. Hyderabad, India, 28 March – 01 April 2011. New York, NY, USA: ACM.
- Zarella, D., 2009. Hubspot. State of the Twittersphere. [pdf] Cambridge, MA: Hubspot.com. Available at: <<http://cdn2.hubspot.net/hub/53/blog/sotwitter09.pdf>> [Accessed 14 March 2015].



**Background questionnaire****Age:** \_\_\_\_\_**Gender:** [ ] Male [ ] Female**Model of mobile phone:** \_\_\_\_\_**Occupation:**

- [ ] Entrepreneur
- [ ] Employee
- [ ] Student
- [ ] Retired
- [ ] Unemployed or on leave

**Education:**

- [ ] Comprehensive or elementary school
- [ ] High school
- [ ] College / University degree
- [ ] Else: \_\_\_\_\_

**Academic and/or working field:**

- [ ] Agriculture
- [ ] Architecture and design
- [ ] Business
- [ ] Divinity
- [ ] Education
- [ ] Engineering
- [ ] Environmental studies and forestry
- [ ] Family and consumer science
- [ ] Human physical performance and recreation
- [ ] Journalism, media studies and communication
- [ ] Law
- [ ] Library and museum studies
- [ ] Medicine
- [ ] Military sciences
- [ ] Public administration
- [ ] Social work
- [ ] Transportation

## Computer and Internet Use

### How do you evaluate your computer skills?

- Excellent, I understand how computers function
- Good, I use computers often and fluently
- I can use basic functions such as email
- I am a novice in computer use
- I don't use computers at all

### How often do you use the Internet?

- Daily or nearly daily
- Few of times a week
- Few times a month
- More rarely than few times a month
- Never

### How often do you use the Internet in your mobile phone?

- Daily or nearly daily
- Few of times a week
- More rarely than few times a month
- Never

### For which purposes do you use mobile phone Internet?

- Sharing links
- Playing music
- Reading and sending email
- Social networks
- Reading the news
- Playing games

Other \_\_\_\_\_

**Use of the Twitter****Have you used Twitter before this test?**

- Yes
- No

**How often do you use Twitter usually?**

- Daily
- Several times during the week, but not daily
- About once a week
- Couple times in a month

**How long time ago have you started using Twitter?**

- Less than month
- Few months ago
- Few years ago

**Which mobile application do you use for Twitter?**

- Official
- Other, \_\_\_\_\_

**How often do you use Twitter.com comparing to mobile application?**

- Using it more often than mobile application
- Using it approximately the same time as mobile application
- Using it more rarely than mobile application
- Other, \_\_\_\_\_

**What do you prefer in everyday usage?**

- Mobile application
- Twitter.com
- No preference

**Do you use Favorite button in Twitter and for which purpose?**

- Yes, \_\_\_\_\_
- No

**Consent form**

We ask you to participate in a usability test that is part of Master thesis work at the University of Tampere. By participating in the usability test you will help to evaluate the usability of the rich content sharing the application for mobile phone

You will be asked to perform different tasks using the service and to think out loud while doing the tasks. In addition, we will ask you to fill in questionnaires and we will interview you about the use of the application. The test will be recorded.

During the test, the audio version of conversation will be recorded. The application will read part of the participant's Twitter timeline only for the purpose of showing them to the participant during evaluation. The data is not stored in permanent storage and will be deleted from mobile phone memory straight after testing. The materials recorded during the test will be used to evaluate the usability of the service in Master thesis work. The recordings will be used only for evaluation and not for anything other than that.

The results of the test will be reported anonymously.

You can stop participating in the usability test at any point.

**By signing this form, you will accept the above terms.**

Date and place: \_\_\_\_\_

Signature: \_\_\_\_\_

Name clarification: \_\_\_\_\_

## **Test plan**

### **Participants:**

Active users of Twitter, preferably, also at the same time using Instagram

### **Research:**

The participants compare their experience while working with Twitter card in both Twitter official client and the prototype application. Evaluation is held by using “think aloud” method while performing tasks, asking additional questions and filling the user experience evaluation questionnaire AttrakDiff™ with related questions after using official client and after using prototype.

### **Hypothesis:**

The users will benefit from increased number of supported by Twitter card services

### **Preparations before testing:**

1. Set screen timeout for mobile phone in long period, not less than 5 minutes
2. Charge phone
3. Install the application beforehand
4. Open AttrakDiff™ page with ready questionnaire

### **Steps:**

1. The moderator describes himself and his background
2. The moderator shortly introduces the prototype application
3. The moderator explains the rules, “think aloud” method, right of the participant to leave testing in any moment, refuse from making tasks he does not like. Explaining that possible data is collected only during testing and will be deleted straight after it under control of the participant
4. The participant signs consent form
5. The participant fills initial background questionnaire
6. The moderator switches on audio recording
7. The moderator prepares a paper and pen, since the participant will need it to remember authorization code from Twitter
8. The participant is testing Twitter card in official Twitter client and perform tasks.
9. The participant fills AttrakDiff™ questionnaire
10. The participant is testing Twitter card in the prototype application and perform tasks.
11. The participant fills AttrakDiff™ questionnaire

12. The moderator deletes the application prototype from mobile phone in order to delete the entire private participant's data from mobile phone. The moderator notifies the participant about this. The moderator leads user to Twitter official page to disable the application access to user's data

**Tasks and questions from the moderator:**

1. "Can you please authorize to the application using your Twitter application using your credentials".
2. "Please find any tweet that has a preview in it"
3. "Do you feel that this tweet is interesting for you, what attracts you"
4. "Do you feel that preview engaged you to interact with tweet?"
5. "Do you feel that preview is enough to "favorite" tweet without going directly to link"
6. "Please find any tweet that does not have a preview in it and remember it"
7. "Fill the AttrakDiff™ form about Twitter card in the context of official Twitter client"
8. "Please find the tweet that had a preview in the official application"
9. "Does it look different from how it looked like in the official application? What do you like or not"
10. "Please find the tweet that did not have a preview in the official application"
11. "Does it look different from how it looked like in the official application? What do you like or not"
12. "Can you please find how you could see which messages you favorited today and earlier?" The participant should move to Likes tab.
13. "Do you feel this tab can be useful in everyday usage of this application?"
14. "Check if any of content can be unliked by you and why"
15. "Can you check which content is popular in the world". The participant should move to Discover tab.
16. "Do you find this tab useful?"

**Additional questions for test of the extended version of the application**

1. "Imagine that this a timeline where you share links with rich content like web sites, videos, images, music and so on. Can you please try now to find some rich content that attracted your attention and you would like to preview it"
2. "Do you understand how Like works by heart in this application?"
3. "Can you find some content from you timeline for which you could give a like"
4. "Do you feel that it is useful to have like function in such application"
5. "Can you please find how you could see which messages you liked today and earlier?" The participant should move to Likes tab.
6. "Do you feel this tab can be useful in everyday usage of this application?"
7. "Check if any of content can be unliked by you and why"
8. "Can you check which content is popular not among your friends, but in the world"
9. "Do you find this tab useful?"

## Summary transcription of user testing

### 1. Participant 1

The first participant thought that the Twitter card previews encourage him to check the tweet. A random example from his timeline was relevant enough not to proceed to the link, because it was obvious enough. Tweets from the timeline had few infographics in the moment of the user testing and they were enough for favoriting. However, often tweets did not have enough data and could not be favorited. The participant would prefer not to see the expanded tweets all the time; otherwise the user interface looks overloaded. The suggestion was to add a button to show and hide extra data. The participant mentioned also that he does not understand how Twitter desktop and mobile versions decide when to show preview and when not to do it.

The logo of the site appeared quite often instead of thumbnail and the participant mentioned that this does not help him to understand what this site is about. The participant also said that images attract in general quite much, because text in timeline looks the same. Avatar sometimes looked the same as other avatars in timeline and thus could not be enough to attract attention. The principle when Twitter card supports only trusted source made him feel safer, since they were more moderated, everyone cannot push whatever he wants.

### 2. Participant 2

The next participant said that his behavior in his timeline is often affected by Twitter card. He uses favoriting as the way to bookmark the link and thus thinks that the preview is not enough for doing so. On the other hand, the preview can be useful for reviewing of bookmarks, since it has additional visual information.

Author's avatar is affecting little the participant, it does not bring any additional information to tweet. The previews are often not relevant (for example, images), but without them scrolling through timeline can be even harder. The participant understands content publishers, who have their own logic which preview to show, and do not want to change this behavior (which could happen in the prototype application).

The participant 2 experienced also technical problems – some previews were not generated even if link was in tweet. In addition, Youtube video was shown in extremely tiny format. Vimeo videos had problems with rescaling.

Instagram does not have any preview in Twitter, but the participant would like to have freedom of choice as well. Twitter card restricts number of supported services and according to the participant it makes it safer to browse the timeline, but the extension of this list takes too long and logic of approval is not clear.

### 3. Participant 3

The participant 3 feels that the tweet with blank link and no Twitter card is less engaging than with it. It will be more likely to catch his attention, provided that the picture is relevant. This participant is using the Favorite button as bookmark and that is why he mentioned that the preview in this case is often not enough, since it is important to have a text overview for the link. The preview itself says little about it, but the text combined with the preview is a good idea. The preview sometimes can be not informative enough and not explaining an article, however it can look “pretty” and thus make clicking a link more probable.

The participant 3 found interesting the idea of short abstract from the link. It increases trust in the site, but it is not that beneficial, since usually the users anyway follow people, whom they trust. The links containing full address of service also looked more attractive than links of default shortening service t.co. The participant does not like to open links in the separate window of the browser application, and thus the support of services of this kind would be useful.

The preview image in some cases was poor and not representing the idea from the link and it is better not to show the preview at all than to show the wrong preview. The algorithm was obvious for the participant: it tries to get preview at any cost so quality is not the first priority. Image animation looks rather “cool”; in the Twitter application it is not supported.

### 4. Participant 4

The participant 4 feels that links from his followees are more engaging than any other links. The thumbnail representing the link can be useful, but it is not a feature that he will miss if it will disappear at some point. The possibility to embed Youtube videos in the Twitter application is a useful feature, because it is not necessary to go all the way to the link just to watch the video. He does not understand the logic of Facebook for similar interfaces – when video is opened in the same page and when in the separate window of browser.

According to the participant, it really depends when the preview is enough for understanding what the link is about. However, favoriting as a bookmark is something that can happen even with preview, especially if he is using mobile phone where it is more convenient to do it. The preview, author and text are important, but the avatar is not that important. The participant mentioned that design of the new application lacks separate containers for separate tweets. If there are many people sharing links then timeline size grows too much, but it can be solved if user interface would have an option to hide these previews. It will be important personally for the participant, because he follows only certain people and there is a little amount of spam. The participant liked the idea of embedded Spotify player, but also said that it can be a



situation when his musical taste is different from the person, whom he follows. This participant does not feel that Twitter lacks additional services for Twitter card, because for him messages are more interesting. Since a slow connection can affect behavior, it is hard to wait for the page to load the preview. The participant found disadvantageous for the end user that Instagram links became unsupported by Twitter after the former was bought by Facebook. The participant was not affected directly by it, since he is not using Instagram.

## **5. Participant 5**

The participant 5 said that Twitter card is an interesting feature and often used by him. It is very useful to have previews of YouTube videos and to watch them with no need to open a separate window of the browser.

The participant would find difficult to scroll the timeline if all the links would have the preview, so some ways are needed to limit it. The abstract of the link is a useful feature and it gives the opportunity to understand the main idea of the link before opening it.

Some of the video links have a wrong sizing, which did not fill the screen. However, the majority of services were working. The previews were representing mainly directly the links. However, some of the links were just logotypes of the sites and therefore not very interesting for the participant.

The participant rarely uses his Twitter for writing and he reads mainly news from different newspapers. In this case, the generated thumbnails are helpful and allow previewing the article directly in Twitter.

## **6. Participant 6. The extended prototype discussion**

According to the participant, links look more attractive when they have previews. Usually he is not following links until he is sure what is content about. Some people share “rubbish”, so he does not like many links of his timeline. Video field was for some reason perceived as small. The interactive previews for different services are a good way to rate content before visiting the link, even if he would not open them in other case.

The participant found the application to be close to Flipboard. Links and stories are really attractive, even if thumbnails are not interactive. Usually the process of opening a link in browser from the application can take few seconds and often is annoying. Like button that was represented by a heart-symbol was understandable with no additional instructions given, as favorite link in Twitter. Though the participant mentioned also that it could be as well a Like feature inside of the story. It is enough for the participant to see only preview to “like” the tweet. The participant also mentioned being affected by number of “likes”, the bigger number of “likes”, and the more

probability that it will be “liked” and checked by him. Unlike button is not very useful in the list of “likes”, since the choice will probably remain unchanged any time later. However if “likes” were used as bookmarks, then it can be useful to remove read links from bookmarks. Discover feature from official Twitter can be useful as well in this application.

## Source code

## script.js

```

$(document).ready(function() {
    // checking if it is a device or desktop browser
    var isPhoneGap = (document.location.protocol == "file:");
    if(isPhoneGap) {
        document.addEventListener("deviceready", onDeviceReady,
false);
    }
    else {
        onDeviceReady();
    }
});

$( document ).on( "click", "a", function() {
    // phonegap in android can open links in new window
    // using call to special function of navigator
    var href = $(this).attr('href');
    navigator.app.loadUrl(href, {openExternal : true});

    return false;
});

function onDeviceReady() {
    var Settings = {
        embedly: 'HERE SHOULD BE EMBEDLY KEY',
        cbkey: 'HERE SHOULD BE TWITTER CONSUMER KEY',
        cbsecret: 'HERE SHOULD BE TWITTER SECRET KEY'
    }

    // initializing Codebird Twitter library
    var cb = new Codebird;
    cb.setConsumerKey(Settings.cbkey, Settings.cbsecret);
    cb.setUseProxy(false);

    // Template for image
    var imageTemplate = twig({
        id: "image",
        href: "templates/image.twig.html?"+(new Date()).getTime(),
        async: false
    });

    // Template for tweet
    var tweetTemplate = twig({
        id: "tweets",
        href: "templates/tweet.twig.html?"+(new Date()).getTime(),
        async: false
    });

    // Template for discover page
    var discoverTemplate = twig({
        id: "discover",
        href: "templates/discover.twig.html?"+(new Date()).getTime(),
        async: false
    });

    // Application initialization
    window.App = {

```

```

    Models: {},
    Collections: {},
    Views: {},
    Router: {},
    State: {
      Views: {},
      Menu: null
    }
  };

  // Router for pages of application:
  // welcome, authorizaion, timeline,
  // favorites and discover pages
  App.Router = Backbone.Router.extend({
    routes: {
      "": "index",
      "welcome": "welcome",
      "authorization": "authorization",
      "timeline": "timeline",
      "favorites": "favorites",
      "discover": "discover"
    },
    index: function() {
      this.navigate("welcome");
    },
    welcome: function() {
      App.State.Views.Welcome = new App.Views.Welcome();
      App.State.Views.Welcome.render();
    },
    authorization: function() {
      // Hiding previous views
      for(var key in App.State.Views) {
        var view = App.State.Views[key];
        view.$el.hide();
        view.undelegateEvents();
      }

      App.State.Views.Authorization = new
App.Views.Authorization();
      App.State.Views.Authorization.render();
    },
    timeline: function() {
      // Hiding previous views
      for(var key in App.State.Views) {
        var view = App.State.Views[key];
        view.$el.hide();
        view.undelegateEvents();
      }

      App.State.Views.Timeline = new App.Views.Timeline();
      App.State.Views.Timeline.render('timeline');

      // Initializing menu
      App.State.Views.Menu = new App.Views.Menu();
      App.State.Views.Menu.render();
    },
    favorites: function() {
      // Hiding previous views
      for(var key in App.State.Views) {
        var view = App.State.Views[key];
        view.$el.hide();
        view.undelegateEvents();
      }
    }
  });

```

```

    }
    App.State.Views.Timeline = new App.Views.Timeline();
    App.State.Views.Timeline.render('favorites');

    // Initializing menu
    App.State.Views.Menu = new App.Views.Menu();
    App.State.Views.Menu.render();
  },
  discover: function() {
    // Hiding previous views
    for(var key in App.State.Views) {
      var view = App.State.Views[key];
      view.$el.hide();
      view.undelegateEvents();
    }
    App.State.Views.Discover = new App.Views.Discover();
    App.State.Views.Discover.render();

    // Initializing menu
    App.State.Views.Menu = new App.Views.Menu();
    App.State.Views.Menu.render();
  }
});

var appRouter = new App.Router();

// VIEWS
App.Views.Welcome = Backbone.View.extend({
  el: $('#welcome'),
  events: {
    'click #start': 'goToAuthorization'
  },
  goToAuthorization: function() {
    cb.__call(
      "oauth_requestToken",
      {oauth_callback: "oob"},
      this.onOAuthCall
    );
    appRouter.navigate("authorization", {trigger: true});
  },
  initialize: function(){
    _.bindAll(this, 'onOAuthCall', 'onAuthReply');
  },
  onOAuthCall: function(reply) {
    cb.setToken(reply.oauth_token, reply.oauth_token_secret);
    console.log(reply.oauth_token) // gets the authorize
screen URL
    cb.__call(
      "oauth_authorize",
      {},
      this.onAuthReply
    );
  },
  onAuthReply: function(auth_url) {
    var isPhoneGap = (document.location.protocol == "file:");
    if(isPhoneGap) {
      navigator.app.loadUrl(auth_url, {openExternal :
true});
    }
    else {

```

```

        window.open(auth_url, " blank", "resizable=yes,
scrollbars=yes, " + "titlebar=yes, width=800, height=600");
    }
},
render: function(){
    var self = this;
    $(this.el).show();
}
});

App.Views.Authorization = Backbone.View.extend({
    el: $('#authorization'),
    events: {
        'click #approve-pin': 'approvePin'
    },
    approvePin: function() {
        var pin = $("#pin").val();
        cb.__call(
            "oauth_accessToken",
            {oauth_verifier: pin},
            this.handleApprovedPin
        );
        return false;
    },
    handleApprovedPin: function(reply) {
        cb.setToken(reply.oauth_token, reply.oauth_token_secret);

        appRouter.navigate("timeline", {trigger: true});
    },
    initialize: function(){
        _.bindAll(this, 'handleApprovedPin', 'approvePin');
    },
    render: function(){
        var self = this;
        $(this.el).show();
    }
});

App.Views.Menu = Backbone.View.extend({
    el: $('#menu'),
    events: {
        'click #menu-home': 'timeline',
        'click #menu-likes': 'favorites',
        'click #menu-discover': 'discover',
    },
    timeline: function(event) {
        $(event.currentTarget).tab('show');
        appRouter.navigate("timeline", {trigger: true});
    },
    favorites: function(event) {
        $(event.currentTarget).tab('show');
        appRouter.navigate("favorites", {trigger: true});
    },
    discover: function(event) {
        $(event.currentTarget).tab('show');
        appRouter.navigate("discover", {trigger: true});
    },
    initialize: function(){
        _.bindAll(this, 'timeline', 'favorites', 'discover');
    },
    render: function(){
        console.log(this.el);
    }
});

```

```

        var self = this;
        $(this.el).show();
    }
});

App.Views.Discover = Backbone.View.extend({
    el: $('#discover'),
    render: function() {
        var html = twig({ ref: "discover" }).render();
        $(this.el).html(html);
        $(this.el).show();
    }
});

App.Views.Timeline = Backbone.View.extend({
    el: $('#timeline'),
    events: {
        'click .like-button': 'like'
    },
    like: function(event) {
        var target = event.currentTarget;
        var tweetId = $(target).attr('data-id');
        var isFavorited = parseInt($(target).attr('data-
favorited'));
        var favoriteButton = $(target).find('span');

        if(isFavorited) {
            cb.__call(
                "favorites_destroy",
                {id: tweetId},
                function (reply) {
                    // Tweet was unfavored, gray color back
                    favoriteButton.removeClass('red');
                    $(target).attr('data-favorited', 0);
                    console.log($(target).attr('data-favorited'));
                    var countElement = $('span[data-
id='+tweetId+']');
                    var value = +countElement.text();
                    countElement.text(value-1);
                    console.log(reply);
                }
            );
        }
        else {
            cb.__call(
                "favorites_create",
                {id: tweetId},
                function (reply) {
                    // Tweet was favored, marking it
                    favoriteButton.addClass('red');
                    $(target).attr('data-favorited', 1);
                    var countElement = $('span[data-
id='+tweetId+']');
                    var value = +countElement.text();
                    countElement.text(1+value);
                    console.log(reply);
                }
            );
        }
    },
    initialize: function() {

```

```

    _._bindAll(this, 'getHomeTimeline', 'handleTweets',
'handleEmbedly');
  },
  render: function(type) {
    $(this.el).hide();
    $(this.el).show();

    switch(type) {
      case 'timeline':
        this.getHomeTimeline();
        break;
      case 'favorites':
        this.getFavorites();
        break;
    }
  },
  getHomeTimeline: function() {
    console.log('home timeline');

    var count = 25;
    cb.__call(
      "statuses_homeTimeline",
      "count="+count,
      this.handleTweets
    );
  },
  getFavorites: function() {
    var count = 25;
    cb.__call(
      "favorites_list",
      "count="+count,
      this.handleTweets
    );
  },
  handleTweets: function(tweets) {
    var tweetsHTML = twig({ ref: "tweets" }).render(
      {tweets: tweets});
    $(this.el).html(tweetsHTML);

    var _this = this;
    // Loading content for each of the tweets
    $('<div class="content">').each(function() {
      var url = $(this).attr('data-url');
      var contentElement = this;

      $.embedly.oembed(url, {
        key: Settings.embedly,
        query: {
          maxwidth: Math.floor($(window).width()/4)
        }
      })
      .progress(function(data) {
        _this.handleEmbedly(contentElement, data);
      });
    });
  },
  handleEmbedly: function(contentElement, data) {
    var html;

    if(data.html) {
      // PhoneGap doesnt understand URLs starting with "://"

```



```

        // Replacing it with normal urls with http
        html = data.html.replace("//cdn.embedly.com",
"http://cdn.embedly.com");
    }
    else {
        var imgSrc="";

        if('thumbnail_url' in data) {
            imgSrc = data.thumbnail_url;
        }

        var options = {
            src: imgSrc,
            width: $(window).width(),
            url: data.url,
            description: data.description
        }

        html = twig({ ref: "image" }).render(options);
    }

    $(contentElement).html(html);
}

});

// Starting backbone router
Backbone.history.start();
}

```

## index.html

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="format-detection" content="telephone=no" />
    <!-- WARNING: for iOS 7, remove the width=device-width and
height=device-height attributes. See
https://issues.apache.org/jira/browse/CB-4323 -->
    <meta name="viewport" content="user-scalable=no, initial-
scale=1, maximum-scale=1, minimum-scale=1, width=device-width,
height=device-height, target-densitydpi=medium-dpi" />
    <link rel="stylesheet" href="css/bootstrap.css">
    <link rel="stylesheet" type="text/css" href="css/index.css" />
    <title>Application</title>
  </head>
  <body unselectable="on">
    <!-- Elements of menu -->
    <ul class="nav nav-pills" id="menu">
      <li class="active"><a href="#" id="menu-home">Home</a></li>
      <li><a href="#" id="menu-likes">My likes</a></li>
      <li><a href="#" id="menu-discover">Discover</a></li>
    </ul>

    <!-- Tweets are published here -->
    <div id="timeline">

    </div>

    <!-- Tweets are published here -->
    <div id="favorites">

```

```

</div>

<!-- Start of authorization process -->
<div id="welcome">
  <div class="jumbotron">

    Please press the button below, it will open Twitter site
in your browser.
    Login to your account, save a authorization pin-code that
Twitter will give you
    and come back to continue authorization.

    <button class="btn btn-default btn-lg btn-block"
id="start">Start</button>

  </div>
</div>

<!-- Second step of authorization. Enter pin-code -->
<div id="authorization">
  <div class="jumbotron">
    <form role="form">
      <div class="form-group">
        <label for="pin">Please enter provided Pin-
code</label>
        <input type="text" class="form-control" id="pin"
placeholder="Enter pin">

        <button class="btn btn-default btn-lg"
id="approve-pin">Submit</button>
      </div>
    </form>
  </div>
</div>

<!-- Test content for Discover tab -->
<div id="discover">

</div>

<!-- Additional JavaScript libraries -->
<script type="text/javascript" src="phonegap.js"></script>
<script type="text/javascript" src="js/codebird.js"></script>
<script type="text/javascript" src="js/jquery.js"></script>
<script type="text/javascript"
src="js/jquery.embedly.js"></script>
<script type="text/javascript" src="js/twitter-
text.js"></script>
<script type="text/javascript" src="js/bootstrap.js"></script>
<script type="text/javascript" src="js/twig.js"></script>
<script type="text/javascript" src="js/jstorage.js"></script>
<script type="text/javascript"
src="js/underscore.js"></script>
<script type="text/javascript" src="js/backbone.js"></script>
<script type="text/javascript"
src="js/script.js?3336232"></script>

</body>
</html>

```

**index.css**

```

html {
    height: 100%;
}

body {
    height: 100%;
    -moz-user-select: none;
    -webkit-user-select: none;
    -ms-user-select: none;
    user-select: none;
}

.red {
    color: red;
}

.no-margin {
    margin-bottom: 0;
}

.content {
    overflow: hidden;
}

.videoWrapper {
    width: 100%;
    position: relative;
    padding-bottom: 56.25%; /* 16:9 */
    padding-top: 25px;
    height: 0;
}

.videoWrapper iframe {
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
}

#menu-items, #authorization, #approve, #test-content {
    display: none;
}

#approve-pin {
    width: 100%;
}

```

**discover.twig.html**

```

<nav class="navbar navbar-default no-margin" role="navigation">
    <div class="container-fluid">
        <div class="navbar-header">
            
            @<span id="nick">backlon</span>
            <span id="name">Dieter Bohn</span>
            
        </div><!-- /.navbar-collapse -->
    </div><!-- /.container-fluid -->
</nav>

```

```

<div class="jumbotron">
  <p>
    RT @CaseyNewton: William Gibson tells @thedextriarchy he
    won't write some of his sci-fi ideas 'cause he fears someone will
    build them!
  </p>
</div>
<div class="videoWrapper">
  <iframe width="100%" height="166" scrolling="no" frameborder="no"
  src="https://w.soundcloud.com/player/?url=https%3A//api.soundcloud.com
  /tracks/34019569&color=0066cc"></iframe>
</div>

<nav class="navbar navbar-default no-margin" role="navigation">
  <div class="container-fluid">
    <div class="navbar-header">
      
      @<span id="nick">backlon</span>
      <span id="name">Dieter Bohn</span>
      
    </div><!-- /.navbar-collapse -->
  </div><!-- /.container-fluid -->
</nav>
<div class="jumbotron">
  <p>
    RT @CaseyNewton: William Gibson tells @thedextriarchy he
    won't write some of his sci-fi ideas 'cause he fears someone will
    build them! http...
  </p>
</div>
<div class="videoWrapper">
  <!-- Copy & Pasted from YouTube -->
  <iframe id="ytplayer" type="text/html" width="100%"
  src="http://www.youtube.com/embed/M7lc1UVf-
  VE?autoplay=0&origin=http://example.com"
  frameborder="0"></iframe>
</div>

<nav class="navbar navbar-default no-margin" role="navigation">
  <div class="container-fluid">
    <div class="navbar-header">
      
      @<span id="nick">backlon</span>
      <span id="name">Dieter Bohn</span>
      
    </div><!-- /.navbar-collapse -->
  </div><!-- /.container-fluid -->
</nav>
<div class="jumbotron">
  <p>
    RT @CaseyNewton: William Gibson tells @thedextriarchy he
    won't write some of his sci-fi ideas 'cause he fears someone will
    build them! http...
  </p>
</div>
<div class="videoWrapper">
  <!-- Copy & Pasted from YouTube -->

```

```

    <iframe class="js-resize"
src="http://player.vimeo.com/video/25924530?title=0&byline=0&p
ortrait=0" width="400" height="225" frameborder="0"></iframe>
</div>

<nav class="navbar navbar-default no-margin" role="navigation">
  <div class="container-fluid">
    <div class="navbar-header">
      
      @<span id="nick">backlon</span>
      <span id="name">Dieter Bohn</span>
      
    </div><!-- /.navbar-collapse -->
  </div><!-- /.container-fluid -->
</nav>
<div class="jumbotron">
  <p>
    RT @CaseyNewton: William Gibson tells @thedextriarchy he
won't write some of his sci-fi ideas 'cause he fears someone will
build them! http...
  </p>
</div>
<div class="videoWrapper">
  <iframe
src="https://www.flickr.com/photos/canadianveggie/8379887392/player/"
width="100%" height="75" frameborder="0" allowfullscreen
webkitallowfullscreen mozallowfullscreen oallowfullscreen
msallowfullscreen"></iframe>
</div>

<nav class="navbar navbar-default no-margin" role="navigation">
  <div class="container-fluid">
    <div class="navbar-header">
      
      @<span id="nick">backlon</span>
      <span id="name">Dieter Bohn</span>
      
    </div><!-- /.navbar-collapse -->
  </div><!-- /.container-fluid -->
</nav>
<div class="jumbotron">
  <p>
    RT @CaseyNewton: William Gibson tells @thedextriarchy he
won't write some of his sci-fi ideas 'cause he fears someone will
build them! http...
  </p>
</div>
<div class="videoWrapper">
  <iframe src="http://instagram.com/p/sTKuWAFMpA/embed/"
width="1200" frameborder="0" scrolling="no"
allowtransparency="true"></iframe>
</div>

```

### image.twig.html

```

<img width='{{width}}' src='{{src}}'><div>
{{description}}</div><br>
<a href='{{url}}'>{{url}}</a>

```

**tweet.twig.html**

```

{% for tweet in tweets %}
<div style="clear">&nbsp;</div>
<nav class="navbar navbar-default no-margin" role="navigation">
  <div class="container-fluid">
    <div class="navbar-header">
      
      @<span id="nick">{{tweet.user.screen_name}}</span>
      <span id="name">{{tweet.user.name}}</span>
      
    </div><!-- /.navbar-collapse -->
    </div><!-- /.container-fluid -->
</nav>
<div class="jumbotron">
  <p>{{tweet.text}}</p>
</div>

{% if tweet.entities and tweet.entities.urls and
tweet.entities.urls|length%}
<div class="content" data-url="{{ tweet.entities.urls[0].expanded_url
}}">
  
</div>
{% endif %}
<div class="favorite-box">
  <button type="button" class="btn btn-default btn-lg like-button"
    data-id="{{tweet.id_str}}"
    data-favorited="{%if tweet.favorited %}1{%else%}0{%endif%}">

    <span class="glyphicon glyphicon-heart {%if tweet.favorited
%}red{%endif%}"></span>
  </button>
</div>
<div class="favorite-box">
  <span class="count" data-
id="{{tweet.id_str}}">{{tweet.favorite_count}}</span> liked this
</div>
<div style="clear: both;">&nbsp;</div>

{% endfor %}

```

**oEmbed response**

```
{
  "version": "1.0",
  "type": "video",
  "provider_name": "YouTube",
  "provider_url": "http://youtube.com/",
  "width": 425,
  "height": 344,
  "title": "Amazing Nintendo Facts",
  "author_name": "ZackScott",
  "author_url": "http://www.youtube.com/user/ZackScott",
  "html":
    "<object width=\"425\" height=\"344\">
      <param name=\"movie\"
value=\"http://www.youtube.com/v/M3r2XDceM6A&fs=1\"></param>
      <param name=\"allowFullScreen\" value=\"true\"></param>
      <param name=\"allowsriptaccess\" value=\"always\"></param>
      <embed src=http://www.youtube.com/v/M3r2XDceM6A&fs=1\
type=\"application/x-shockwave-flash\" width=\"425\" height=\"344\"
allowsriptaccess=\"always\" allowfullscreen=\"true\"></embed>
    </object>",
}
```

**Bootstrap**

This is a minimum HTML document needed to start working with Bootstrap in it:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap 101 Template</title>
<!-- Bootstrap -->
<link href="css/bootstrap.min.css" rel="stylesheet">
<!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media
queries -->
<!-- WARNING: Respond.js doesn't work if you view the page via file://
--> <!--[if lt IE 9]>
<script
src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></scrip
t>
<script
src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></sc
ript> <![endif]-->
</head>
<body>
<h1>Hello, world!</h1> <!-- jQuery (necessary for Bootstrap's
JavaScript plugins) -->
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js
">
</script>
<!-- Include all compiled plugins (below), or include individual files
as needed -->
<script src="js/bootstrap.min.js"></script>
</body>
</html>
```

**Bootstrap CSS styles**

```
<!-- Latest compiled and minified CSS -->
```

```

<link rel="stylesheet"
href="//netdna.bootstrapcdn.com/bootstrap/3.1.1/css/bootstrap.min.css"
> <!-- Optional theme -->
<!-- Latest compiled and minified JavaScript -->
<script
src="//netdna.bootstrapcdn.com/bootstrap/3.1.1/js/bootstrap.min.js"></
script>

```

## Navbar

*Navbars are responsive meta components that serve as navigation headers for application or site. They begin collapsed (and are toggleable) in mobile views and become horizontal as the available viewport width increases.*

```

<nav class="navbar navbar-default" role="navigation">
<div class="container-fluid">
  <div class="container-fluid">
    Nav bar content
  </div><!-- /.navbar-collapse -->
</div><!-- /.container-fluid -->
</nav>

```

## Embedly

```

<head>
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"
type="text/javascript"></script>
<script src="http://cdn.embed.ly/jquery.embedly-3.1.1.min.js"
type="text/javascript"></script>
</head>

```