[WORK IN PROGRESS]

Spring | 15

TicText Project Documentation

Terrence Katzenbaer, Kevin Chen, Jack Arendt, Chengkan Huang, Georgy Petukhov

[Type the abstract of the document here. The abstract is typically a short summary of the contents of the document.]

The Goals of the Documentation

You should write your documentation as though you are explaining your project to a new developer who is about to join your team. You should include enough details so that he or she can get started quickly. On the other hand, the documentation should not overwhelm them with too much information that can be easily discovered by reading the code.

Your documentation should include pointers to important facts that you wished you had known when you started the project, e.g., limitations of a framework you used or a workaround for a nasty bug you encountered, etc. These are things that people cannot figure out quickly unless it has been documented.

Deliverables

Submit the documentation along with your code under the **final-release** tag of your code repository.

Final Documentation

The documentation is worth 20 points. It needs to include the following items (you can use this as a guide for the table of contents):

You should be able to re-use material from the homeworks and your writing assignments (if you are taking CS429) while preparing the documentation.

Formatting Guidelines

1. Your document must have a table of contents.
2. All class, method names, code must be in `monospace` font. Use a font like Courier, Monaco, etc.
3. All figures must be labeled.
4. Do not draw the diagrams by hand – use a diagraming tool.
5. All pages must be numbered.
6. Your documentation should follow consistent formatting for sections, headers, etc., for the *entire* document.
7. Your documentation should be **at least** 5 pages but not more than 15 pages.

Failure to follow *any* of the formatting issues will result in severe deductions (possibly resulting in 0).

Please proof read your documentation to avoid spelling mistakes and grammatical errors. Follow the rules that you have learned from CS429 to structure your sentences and paragraphs. **Do not write in a colloquial manner**. In short, make a *professional* document for submission. Note that it *can* include humor and yet be professional.

Keep your documentation short and to the point. If you are using a framework, you don't need to describe the framework in detail. Instead focus on the big picture. For instance, if you are using a web framework, then you should just explain the *metaphor* behind it (eg., "Web Framework X is based on the Model-View-Controller pattern. Our models follow the Active Record pattern to interface with the database. ...").

What is TicText?

Brief Description

TicText is an iOS messaging app that allows you to send expire-able text with your friends. We call this kind of text "Tic", as in the sound of a clock counting down (Tic Tok, Tic Tok...). For each Tic sent to your friends, you will be able to set a time duration, which will be the amount of time before the Tic got expired and becomes unable for your friend to retrieve its content. That is to say, the recipient can only retrieve the content from our server within a certain time period after the sender sent the Tic. Here is an example of a possible scenario of using TicText:

John (drunk, 3 am, send to Mary, expire in 60 minutes): ***Hey babe, I miss you so much and I really regret breaking up with you...***

Mary (sleeping, her phone received a notification): **"Someone has just sent you a Tic! Swipe to read before it's too late. Tic Tok..."**

Mary (next morning, saw the notification, opened the app and the only thing left was): **"Sorry you've missed a Tic which expired on 4:00 am. "** (an in-app purchase of $4.99 will unlock the sender's name :)

TicText would also let you to send a Tic anonymously - even if the recipient saw the notification in time, he/she still won't be able to see the sender's name.

We believe TicText is also capable of creating romance in many ways. But in the example above, well, hopefully TicText just avoid something awkward or even more.

Motivation

We would like to build something brings people joy and more importantly, works even with limited amount of users. So we gave up on the ideas related to sharing, social networking, or dating, and focused on single user or 1-on-1 experiences. I believe TicText could be something we, especially the younger generation, would enjoy. Also, building a mobile app with modern technologies will always be challenging and exciting. From my past experiences, I'm sure we could learn a lot and have fun as well.

Risks/Challenges

One of the biggest challenges would be building a good-looking UI with great user experience. There wouldn't be many features in TicText so UX will be the top priority with no doubts. In the meantime, we have to think about the security of all the Tics as well. The last thing we want is compromising user's privacy. In order to do that, we need more than beautiful code, but also coding in the "right way" using the "right tools". Another possible challenge is learning how to code and deploy on Parse if we choose to use it as our cloud platform.

Formal Software Development Process

The process followed by your team (make use of your essay if you are taking CS429) - (5 points)

Make sure to address the issues of iterative development, refactoring, testing and collaborative development (even if you are not using XP, you have to address these issues in your documentation).

Requirements & Specifications

Requirements & Specifications of your project (make use of HW2 and HW3) - (5 points)
This should include the user stories and use cases which you implemented in your project. You do not need to mention the requirements which were dropped.

Architecture & Design

(make use of HW2 and HW3) - (5 points)

1. Include multiple UML diagrams to show the important parts of your system (you must have UML diagrams).
2. Describe in a top-down manner the architecture of your system.
3. Include enough details about the design of your system such that anyone who refers to your documentation can understand the major components of your system and how they are related.
4. Describe how the choice of the framework influence the design of your system.

Future Plans

Future plans - (3 points)

Include personal reflections on the project and process by each team member. Describe what you learned.