

2QUEUEOOL4U

OCTOBER 31, 2017

---

PROTOTYPE DEMO



## Metadata:

Project name: NextText

Team name: 2QueueOOL4U

Team members:

James Mulvenna/100965629

Yue Zhang/100980408

Devon Plouffe/100715712

## Documentation:

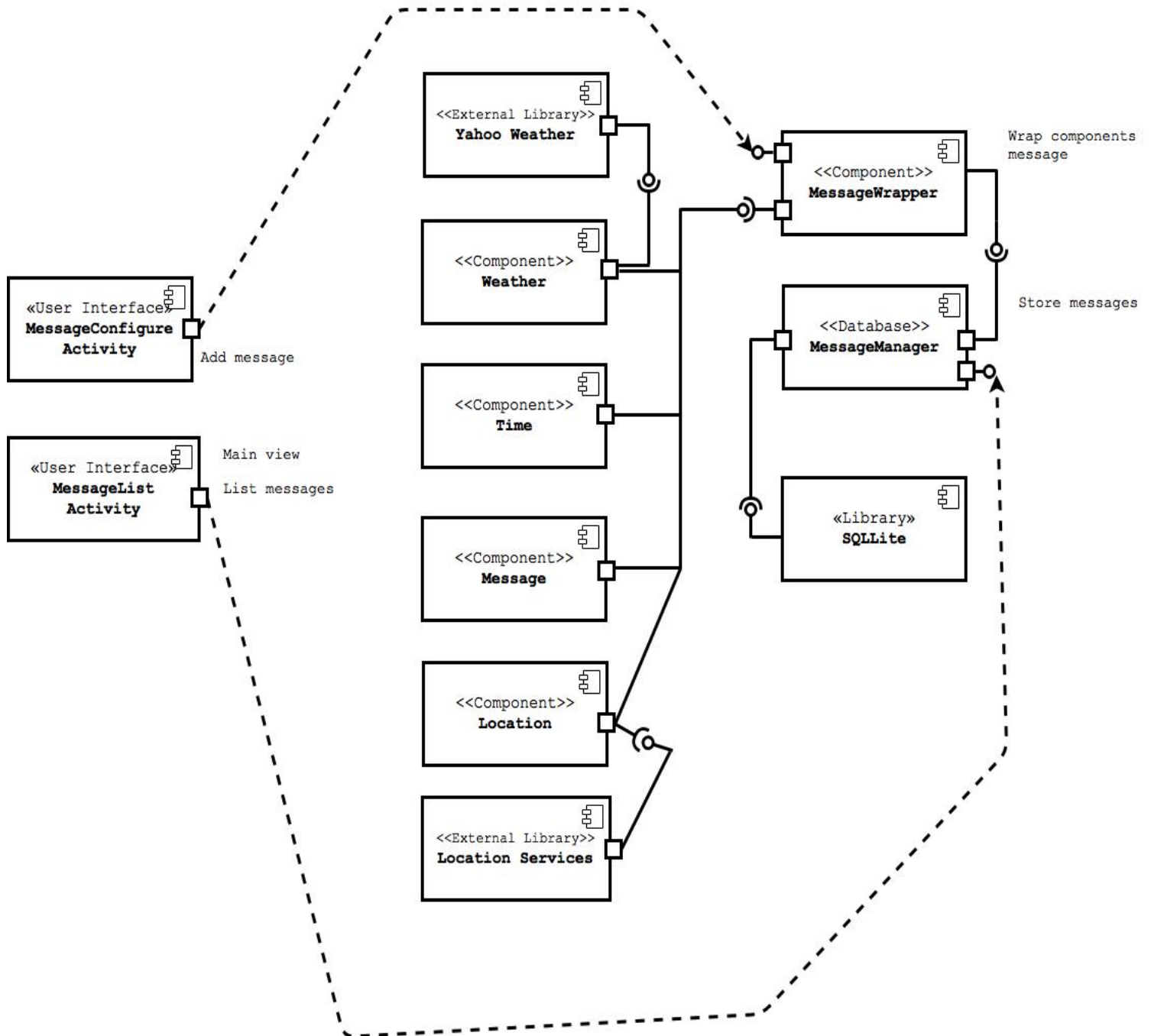
### Demo Summary:

The 2QueueOOL4U demo will include no simulated data. At this time, we have currently developed the entire backend (database) and models making use of their functionality with regards to the frontend. From a developer's point of view, the "NextText" application is backend heavy. Data must be able to flow smoothly in and out of the database with no faults. At this point in the project, we have made the user experience abstract and functional. The user is able to add a SMS or Email with the mandatory fields respectively, they are able to pick a time trigger which includes both a date and time. When the Message object is completed it is added to the database. Following, the Main Activity (frontend) will pull from the database and display to the user all currently queued messages tapping into and displaying useful information such as who the message is to and the content it contains. The user is able to sort their queued messages based on their type (SMS, Email) or all. Finally, they are also able to delete selected messages or all. It is essential to repeat; all of these actions will be on **real** data that a user may apply.

## Report:

2QueueOOL4U has now successfully implemented the backend and made use of its functionality abstractly in the user space. The user space supports sorting messages, adding complete messages, deleting selected messages or all messages. The backend is developed based on SQL using the SQLite library Android provides to its developers. It currently serves methods such as create table, get, get all, add, delete, delete all, count and more. We believe it was essential to have this component of the application done above all to sustain data flow and integrity. We faced many challenges while testing and pushing our database to the limits. We spent endless time modifying the database to fit our software requirements. Drawn-out lengths were allotted to decide which models were most appropriate to serve the database and user experience. Another major challenge we are currently facing is understanding Android development and its services applicable to our project scope, as none of us have created an application to this extent prior to COMP 3004. The learning curve is steep however we have been successful thus far. Outlining the next month of development, 2QueueOOL4U will be implementing the sending service (send SMS, Email) for the time trigger with the use of Android's AlarmManager library which will conclude one user scenario. Continuing, we will have a better understanding of the sending service and carry out the location trigger using the Google Maps API and geo-fencing operations. Finally, we will use the Yahoo Weather API to create the final trigger, weather. If planned testing/debugging time is improbable during progression in any of these components we will sacrifice a trigger in order to span time to test and debug cases as we feel this is a crucial element of any software development. We value a working project in its fulfillment over a project that supports all cases in an imperfect sense.

## Component Diagram:



Ball and socket: Provided component interface (socket), required component interface (ball)

Dotted line + arrow and ball: The dotted line + arrow component uses the ball component

Physical devices supported: Android phone products