Function Decomposition:

In order to build our product, we decomposed our design process as 4 main functionality blocks: UI & Frontend block; image processing and word recognizing block; database generation and word searching block and finally network extension block.

Step 1: When user opened the application, the application will open the camera and let the user tap the word he or she wants to search. The Frontend will then cut the word out and send it to image processing and word recognizing block.

Step 2: The image processing and word-searching block will probe the word letter by letter. For each letter it will compare the image and calculate the similarity against the standard image of 26-alphabit letters. It will then send the recognized word to database searching block.

Step 3: The database-searching block will query the database and retrieve out the meaning of that word and display it on the screen. After that, it will hand the process off to network extension block.

Step 4: The network extension block will invoke the web browser and search on the Internet.

Concept Generation:

We have generated several concepts. We will list it below:

For Frontend, we have several concepts: freeze-and-select, tap-and-translate, frame-aided translation and auto-hiding extensions.

**Freeze-and-select** means user will first tap the screen to freeze the current image and select it on the phone. We use freeze-and-select because we believe this will save the user from holding still the phone for all the time during translation and this will also reduce the iteration of image processing.

**Tap-and-translate** means the user will tap the screen on selected word and the translation will be depicted on the screen.

**Frame-aided translation** is a compromise for speed because if the whole image were frame free, the program would need to deal with a huge more possibility of image distortion. We introduce the frame to regulate user’s behavior in order to provide smooth using experience.

**Auto-hiding extension** means we the network extension button will be hided usually and will show up only after user tapped on the “show extension button”.

For image processing we have several concepts: letter segmentation, standard-letter-image comparing, distortion detection and characteristic grabbing.

**Letter segmentation** will cut out each letter from a word. Basically what we do is checking the pixel density horizontal direction. Because of there is little space between each letter, and the pixel density for a space is 0. So we will use those 0 pixel density as barriers for a letter and cut out the letter image between 2 barriers.

**Standard-letter-image comparing** is how we going to recognize the letter from a letter image. The basic idea is we will compare the image against the library of saved standard image of all 26 letters and choose the most relative one.

**Distortion detection** is the way we detect and restore distortion as user may swing the camera while freezing the screen shot. We will recover the image by saving some distorted standard letter image in training data.

**Characteristic grabbing** will make letter recognition easier as each letter has its own characteristics. We will recognize letter much easier by starting at comparing the characteristics of a letter and it may save lots of computation recourses.

For Database searching we have several concepts as well: mixed searching and history cache.

**Mixed searching** is a mixed searching scheme of target word. First of all we will use first letter as index letter and after that we will use the second letter as sub-index. This will continue either when the tree structure reach to a leaf or reach to 5th level. After that we will use linear search. By this we reduced searching time while introducing acceptable storage overhead.

**History Cache** is a cache storing recently stored word. We assume when user reading scientific document he or she may encounter a new word several times while reading. Thus we will first do a linear or binary searching in history and only when we fail to find the word we went to normal searching scheme.

For Internet extension, we have only 1 concept that is multiple Incoming Resources.

**Multiple Incoming Resources** determines how many website extensions we have. For now, we are planning to have Wikipedia, IEEE Xplore, ACM Library.

Concept Selection Process

We select Concepts by first valuing it against user experience. For example, all of the concepts explained above in Frontend section are very important for user experience. We do not want the user to hold the cell phone still for a long time so we have freeze functionality; we do not want the user to type in the word and this is the whole reason for this project, and we also want to shorten searching time so we use mixed searching scheme. Overall, from the point of user experience, we choose: freeze-and-select, tap-and-translate, auto-hiding extensions, mixed searching, history cache and multiple Incoming Resources.

However, user experience is not the only rule. We also need to take response time into consideration and sometimes we need to make compromise with user-interface. Such as Frame-aided translation, introducing frame will discourage user because they need to fit the paragraph into the frame, but this will largely reduce the time to figure out the font size and the picture direction, it will also reduce time from turning and twisting the picture to make it aligned. From this consideration, we choose: Frame-aided translation, mixed searching and history cache.

Finally, we need to take precision into consideration. We wish our application would be fast as well as accurate. Mostly all of the concepts in image processing section fit into this area but also frame-aided translation as well. So from this consideration we choose: letter segmentation, standard-letter-image comparing, distortion detection, characteristic grabbing and frame-aided translation.

Prototype Description

As our project is pure software project, we won’t be able to manufacture the product. However, as we are building this project in a learning process, we do not have enough time or knowledge to make it perfect. We basically show out the idea of input free searching.

One thing we don’t have in our prototype but may be included and further developed is a full and powerful word database. For now constructing such a big 5 level tree style database is impossible even and importing existing word database into our own is a pure labor work. But if the product were finalized, we would have a whole word database.

Another thing we do not have for now is frame free. Frame free recognizing require smart detection of text direction and also zooming the picture. This will need more training data and constructing training data is another labor work, as we need to manually input all kinds of distorted and stretched letter image.

Other then this two, the prototype should be the same with finally product and we believe the prototype can perfectly represent our final product.