**Vision Document for WhatsApp-Spring**

**Team members:**

*987175 Augusto Gonzalez*

*986970 Ammanuel Desta*

*986988 Jean Vilaire Octavius*

*986226 Saw Qua Lar*

**1. Introduction**

An Instant Messenger is an application found in virtually any phone/computer. Today, users have a lot of choices: Facebook Messenger, WhatsApp, WeChat, Telegram, etc. These applications are simple but offer a lot of functionality. We think it's a good way to get familiar with the Spring framework.

Several years ago in the Computer Professional MS in CS program, there were three

entries per year and student entry numbers were 20-40 per entry. Often there was just one

elective class being offered per block and all students in an entry took the same classes in

the same sequence. Scheduling of classes and faculty was done with a relatively simple

Excel spreadsheet and students were assigned to classes via a manual process.

As Compro has grown, we now offer 4 entries per year and there are often 100 – 130

students per entry. In some blocks, we may offer 8 or 9 elective classes, plus there are

often 3 FPP classes and 5 MPP classes offered per entry. There are several areas of

specialization for classes such as:

- Web Applications

- Data Science

- SW Design

- Networking

- Operating Systems

- Compilers

- Parallel Programming, etc.

Most faculties have one or two areas of specialization and a set of classes that they

would like to teach. In addition, they have preferences for what blocks they can teach.

Faculty needs to be able to enter their profile and be able to view their scheduled classes.

Compro students should be able to view the schedule and register for classes.

A few 500 level courses have 400 level **course prerequisites**, so the 400 level courses

should be offered for each entry in their first blocks on campus.

The 500 level classes should be provided for their later blocks on campus.

Most students take 4 elective blocks on campus.

Some U.S. resident students take 9 elective blocks on campus.

Some OPT students take 5 courses on campus.

MUMSched is a new software tool that will build a Compro schedule of classes with

faculty assigned to each class and will also offer a simple tool for students to register for

those classes.\*

(\*Note – the student registration part will be kept simple for our project. It is added for

the purposes of having a separate student register subsystem – to be explained in later.)

**2. Positioning**

**2.1 Problem Statement**

*[Provide a statement summarizing the problem being solved by this project. The following format may be*

*used:]*

|  |  |
| --- | --- |
| The problem of | *allowing instant digital messaging among users* |
| Affects | *common users* |
| the impact of which is | *rich information exchange is complex, users expect fast communication and ease of use* |
| a successful solution would be | *a tool which sends different types of messages (text, audio, video). This tool will provide a Database to keep track of Message History and will provide a user-friendly interface.* |

**2.2 Product Position Statement**

*[Provide an overall statement summarizing, at the highest level, the unique position the product intends to*

*fill in the marketplace. The following format may be used:]*

|  |  |
| --- | --- |
| For | *Any kind of person* |
| Who | *Everybody that can use a phone/computer.* |
| The WhatsApp-Spring | *is an instant messaging application* |
| That | *allows instant communication between people.* |
| Unlike | *the original WhatsApp* |
| Our product | *is built on the Spring Framework* |

*[A product position statement communicates the intent of the application and the importance of the project*

*to all concerned personnel.]*

**3. Stakeholder Descriptions**

**3.1 Stakeholder Summary**

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Responsibilities** |
| Users | Send/Receive messages | Users are responsible for signing in to the application, adding/ removing contacts and sending/receiving messages. |
|  |  |  |
| Developers | All team members will take part in building the project. | Build the system given Use Case Requirements |
| Testers | All team members will take part in the testing phase. | Write and Run tests |

**3.2 User Environment**

*[Detail the working environment of the target user. Here are some suggestions:*

*Number of people involved in completing the task? Is this changing?*

*How long is a task cycle? Amount of time spent in each activity? Is this changing?*

*Any unique environmental constraints: mobile, outdoors, in-flight, and so on?*

*Which system platforms are in use today? Future platforms?*

*What other applications are in use? Does your application need to integrate with them?*

*This is where extracts from the Business Model could be included to outline the task and roles involved,*

*and so on.]*

Application target use is individual. Which a person should be able to operate it by itself. Users are expected to use it sporadically several times a day. Conversations may take a couple of seconds to several minutes.

We are currently targeting a Web Interface but also mobile and Desktop are possible.

**4. Product Overview**

**4.1 Product Perspective**

The released app will be self-sufficient and completely dependent on itself to carry out all the highlighted functionality. It will have all the features mentioned above and it will be very easy to use.

*[This subsection of the* ***Vision*** *document puts the product in perspective to other related products and the*

*user’s environment. If the product is independent and totally self-contained, state it here. If the product is a*

*component of a larger system, then this subsection needs to relate how these systems interact and needs to*

*identify the relevant interfaces between the systems. One easy way to display the major components of the*

*larger system, interconnections, and external interfaces is with a block diagram.]*

**4.2 Assumptions and Dependencies**

We have made the following assumptions:

The only assumption is to use a modern Web Browser with Javascript Capability.

*[List each factor that affects the features stated in the* ***Vision*** *document. List assumptions that, if changed,*

*will alter the* ***Vision*** *document. For example, an assumption may state that a specific operating system will*

*be available for the hardware designated for the software product. If the operating system is not available,*

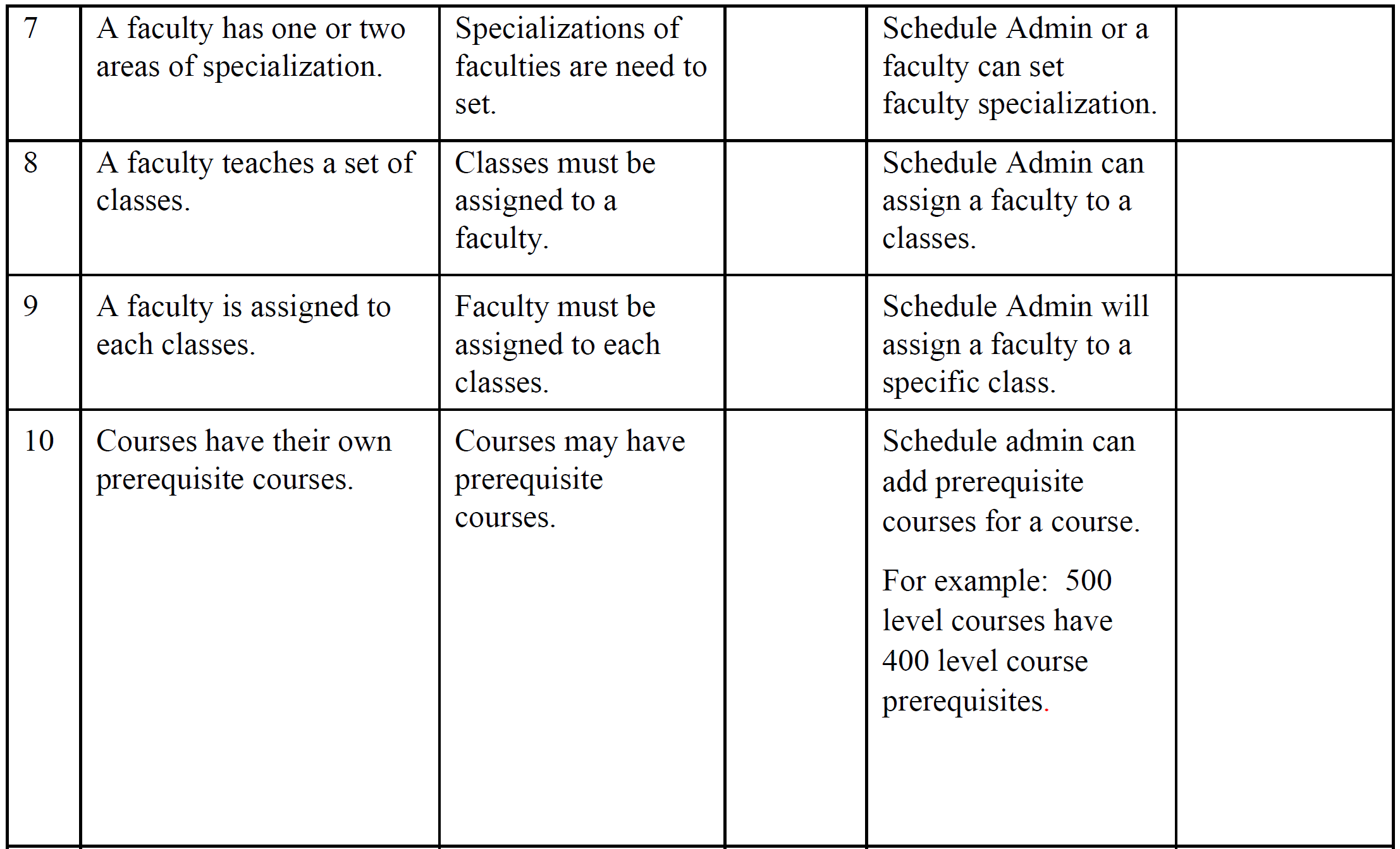
*the* ***Vision*** *document will need to change.]*

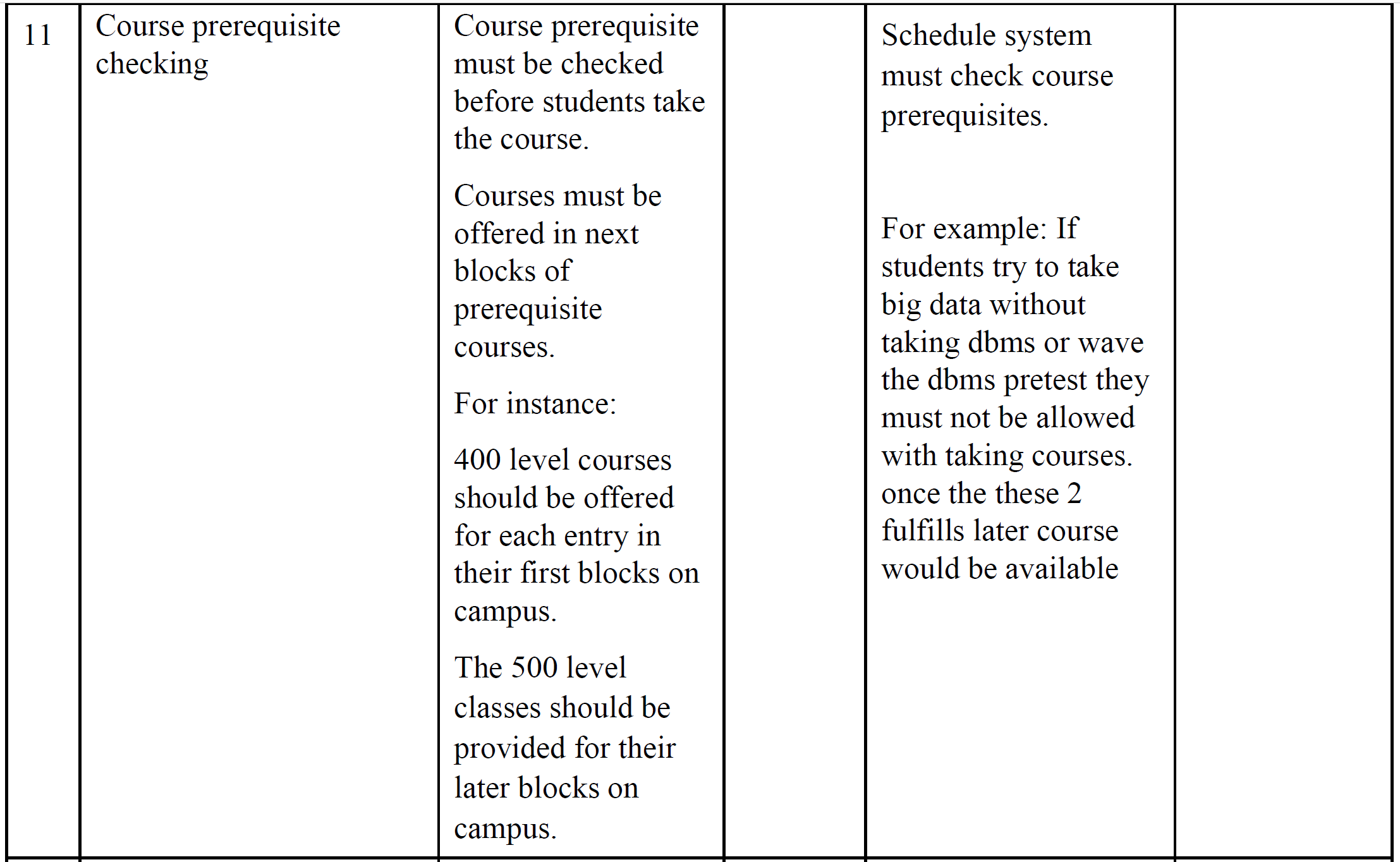
**4.3 Needs and Features**

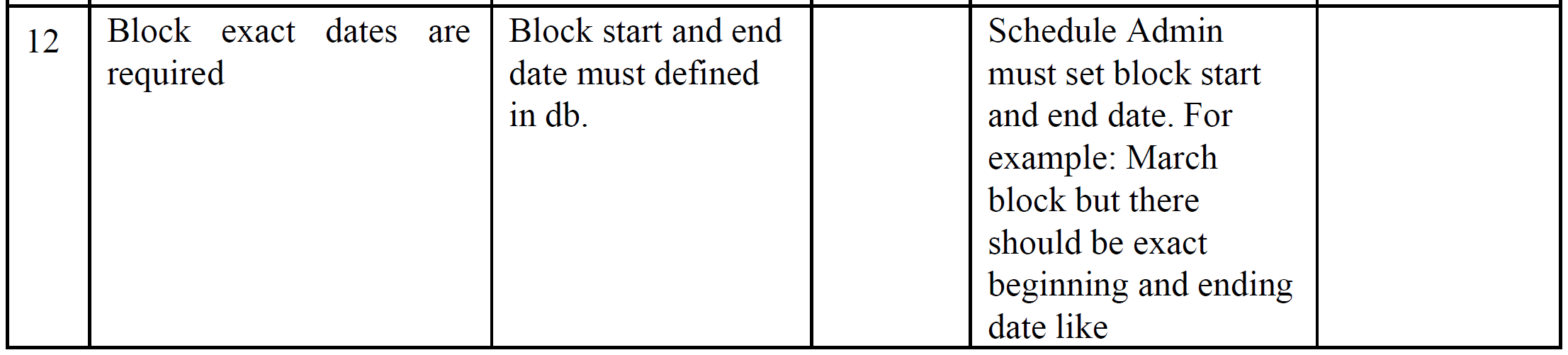
*[Avoid design. Keep feature descriptions at a general level. Focus on capabilities needed and why (not*

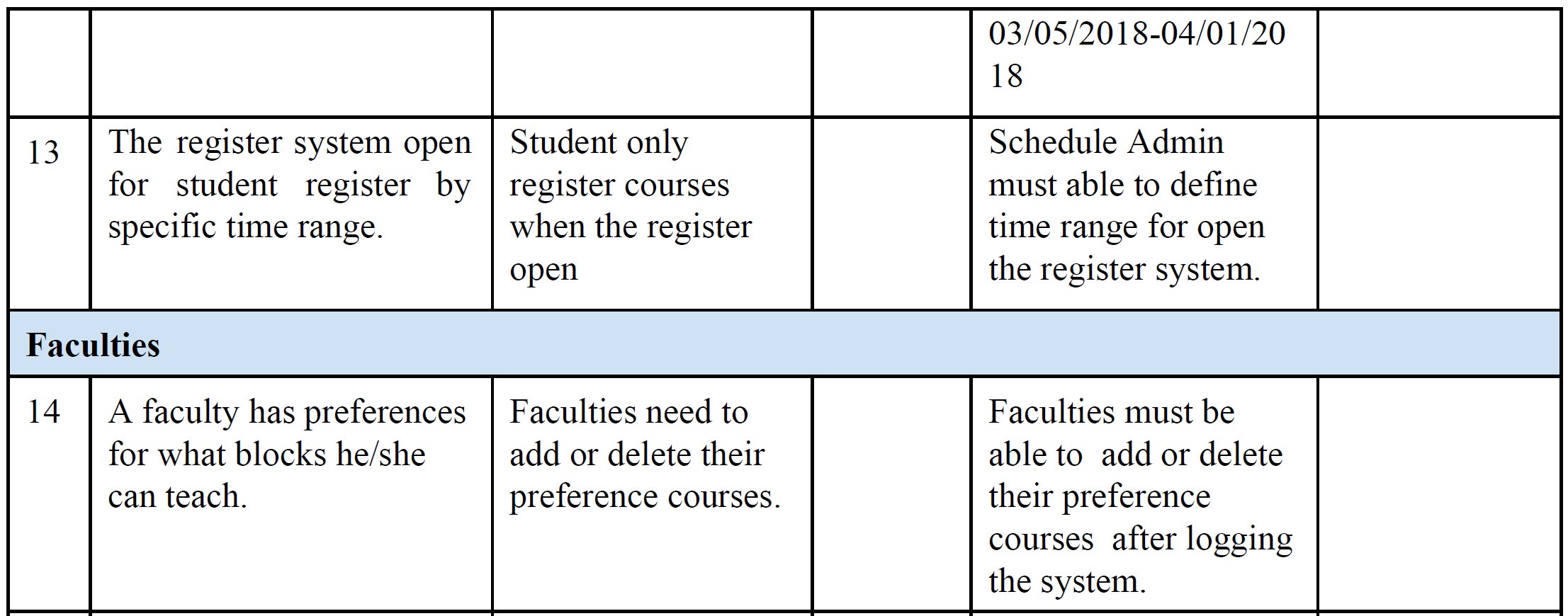
*how) they should be implemented.]*

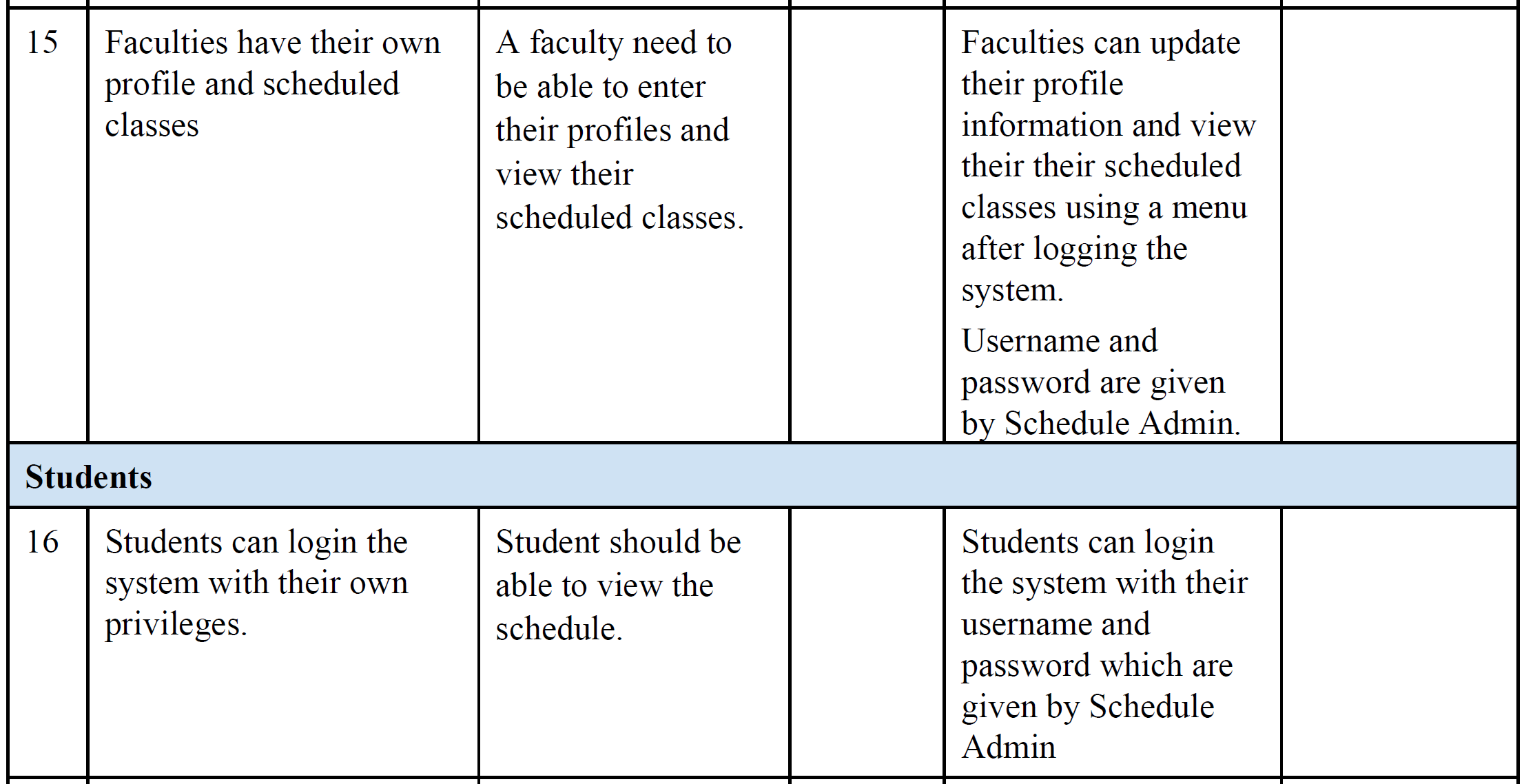
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Problem | Need | Priority | Features | Planned Release |
| User | | | | | |
| 1 | Long distant communication used to be slow and expensive. | Users need faster ways to get in touch. | High | Users should be able to exchange many types of information (text, audio, video) fast and reliably. | 1st-2nd week |
| 2 | Privacy and security are very important to users. | Store Message History safely and securely. | High | Information exchanged should be visible only to the conversation owners. Need to implement a Sign-in module. | 1st-2nd week |
| 3 | Conversations get messy with time and | Users need to keep their conversations organized and have control over their network of contacts. | High | Users should be able to manage their contact list. | 1st-2nd week |
| 4 | A user's network can grow very big. | Users want control over who can contact them. | Medium | Users should be able to block other users. | 3rd-week |

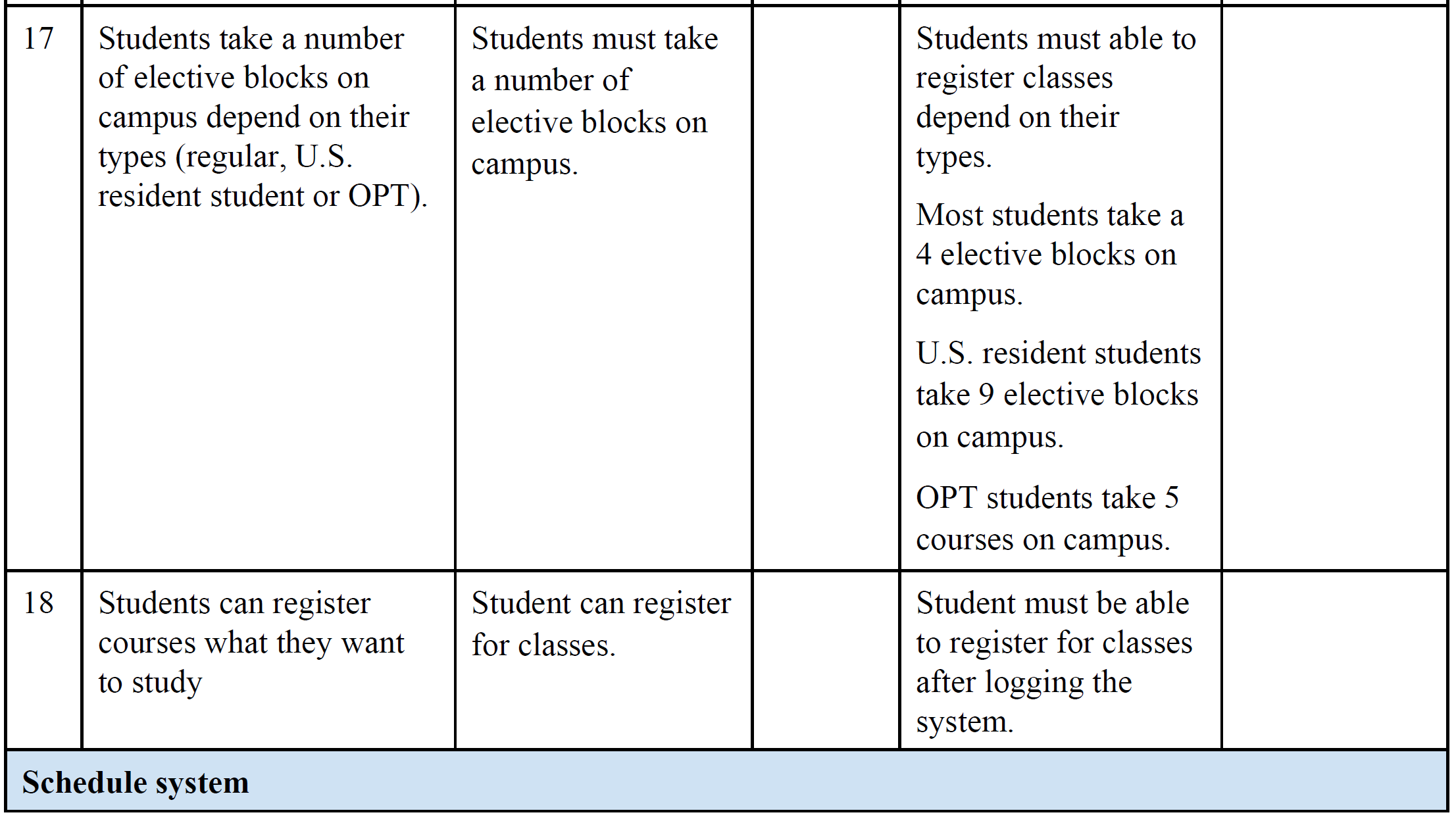


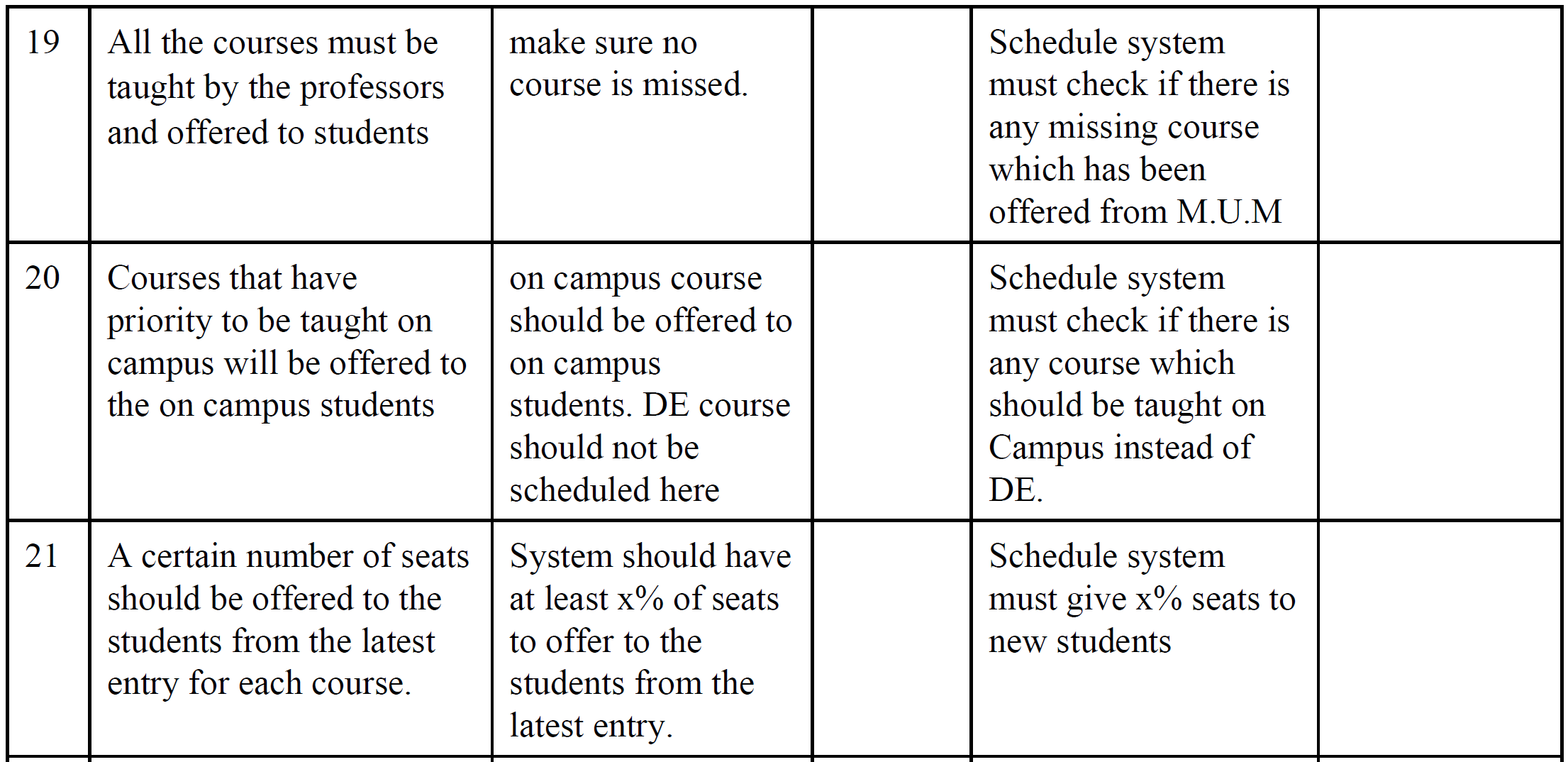


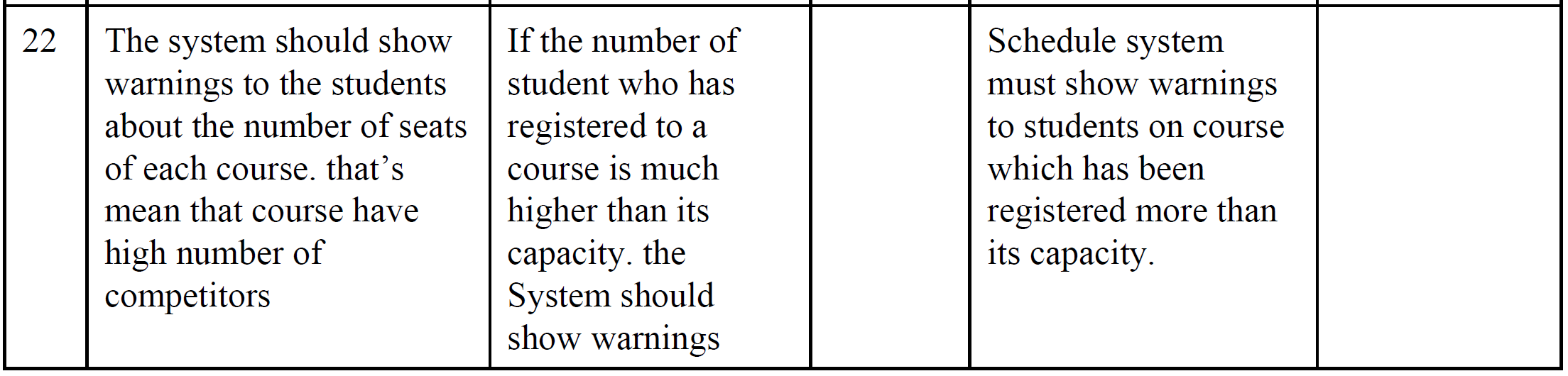












**4.4 Alternatives and Competition**

*[Identify alternatives the stakeholder perceives as available. These can include buying a competitor’s*

*product, building a homegrown solution, or simply maintaining the status quo. List any known competitive*

*choices that exist or may become available. Include the major strengths and weaknesses of each competitor*

*as perceived by the stakeholder or end user.]*

Facebook Messenger, Telegram, WeChat, SMS Messaging

**5. Other Product Requirements**

*[At a high level, list applicable standards, hardware, or platform requirements; performance requirements;*

*and environmental requirements.*

*Define the quality ranges for performance, robustness, fault tolerance, usability, and similar*

*characteristics that are not captured in the Feature Set.*

*Note any design constraints, external constraints, or other dependencies.*

*Define any specific documentation requirements, including user manuals, online help, installation,*

*labeling, and packaging requirements.*

*Define the priority of these other product requirements. Include, if useful, attributes such as stability,*

*benefit, effort, and risk.]*

The application should work seamlessly on new and old computers. The only requirement is a browser. A keyboard is also a requirement because we don't intend to implement Text-To-Speech functionality.