Interpretations of requirements:

* As specified in the brief, the system has a number of different types of users which log in to the system. In addition to this, we have implemented logging out of the system so a different user can log in without having to exit the system.
* The system has four different types of user: Admins, Registrars, Teachers and Students. So, when the brief says that an admin can ‘add and remove user accounts’, we have interpreted this to mean that an admin can add all four different types of users. This means both admins and registrars can add students.
* The grades for retaken examinations are stored in the database out of the full 100 marks and only capped when calculating the mean.
* The beginning of a student email address consists of a concatenation of the initial letter of their forename, their surname and a unique distinguishing integer as stated in the brief. This concatenation will also be their unique username.
* A 4-year integrated masters degree will be considered undergraduate whereas a 1-year masters programme will be considered postgraduate.
* If a degree has credits that aren’t taken up by core modules, the registrar will add on optional modules that will be selected from a list depending on the period and level of study. A student will not be able to choose free modules.
* The system supports up to 26 periods of study but we have assumed that a student won’t reach or exceed that number as failing twice means they cannot proceed.
* Each student has a tutor stored in their information, we thought this could be linked to the teacher user accounts but the brief says to store it as a single string so it is not.
* The brief states that modules have a default number of credits based on their level, however we have interpreted this as the admin would know the default system so will just input the number of credits.
* Any module added after a student has been added will not affect that student as, for example, if a student is in first year and the admin wants to add a different module for the next first-year students it shouldn’t be added to a current student. Therefore, all modules for year must be added before the students are.
* The system includes both deleting modules and a deleting module links as if a module is deleted (for example removed from the first-year course) it cascades to delete the grades for a student which may be needed to calculate the final grades. The deleting module links allows the module to no longer be on a certain degree but previous grades to still exist and the module to still exist on other degrees. If a degree or department is deleted, it would cascade to delete grades too but we haven’t implemented deleting links in these cases as we decided if you were deleting a degree or department you wouldn’t need any further information from them.
* The brief states that “if they fail a 4-year degree at level 4, then they must graduate with the equivalent bachelor’s degree with credits already obtained. From this we have interpreted that for every 4-year integrated masters degree there is a 3-year bachelors equivalent so that if the 4th year is failed the 3rd year grades are used and they are awarded the bachelor equivalent. We also decided that any credits obtained from level 4 wouldn’t count towards the degree if they failed level 4 even if they have already been obtained.
* The grades for modules already passed are transferred to the new period and they retake modules (just once) whose grades are capped.
* 1-year MSc courses count as a conceded pass if a module other than 60 credits dissertation is marginally failed and the pass mark is 50%.

Security features:

* We store a hash of the password using the PBKDF2 algorithm, with HMAC as the pseudorandom function and SHA-1 to generate a fixed-length digest (all done automatically by Java). We store each password in the form salt + ‘$’ + hash(password, salt). Salts are added to negate the effectiveness of rainbow table attacks. An attacker with our codebase/database wouldn’t be able to precompute hundred of potential hashes using the same algorithm as each one has a 16 bit salt. For the purpose of our application we have set the number of iterations to 1,000 as was recommended in 2000 but since computers are more powerful we would likely increase this to make the computation slower.
* A minimum password length has been implemented to reduce the effectiveness of a brute force attack as longer passwords have more possible values therefore increasing the time taken for the brute force attack. We considered forcing users to use a mix of upper and lower case or special characters but security experts disagree on the validity of such a method.
* Throughout the application, wherever a password or any confidential piece of information is entered, it entered into a JpasswordField and stored as a `char[]` as since Java 8, String can be automatically interned and stored on the heap. Whilst storing it as a char array only serves to reduce the window of opportunity that an attacker looking to dump the memory and find this vital information has, it is still a security recommendation
* A password field has been used when a user logs into the system which obscures the password, ensuring log in details cannot be obtained by watching someone log in.
* Where possible in data input, we have used drop-down lists so the user can only choose from what is already in the database. For example, when creating a new degree, the lead department can be chosen from a drop-down list of existing departments as the system would break if they tried to link a degree to a department that didn’t exist. In some cases, text fields had to be used, for example, adding the name of a new department. If these need to be in a specific format they are verified, for example, department code needs to be three capital letters. The database has maximum lengths on a number of string fields, but if the input was longer this would be caught in a try, catch statement so the database wouldn’t break.
* The system manages authorisation through Role-Based Access Control (RBAC), where the four different types of users are the different roles. Admins, Registrars and Teachers can read and write to certain tables of the database based on their tasks. Students can only read from the database, and the data must be related to them otherwise a breach of privacy could occur.
* We have used prepared statements to prevent SQL injections as this allows the code to distinguish between the code and the input regardless of the data inputted (so the user can’t change the intent of a query).