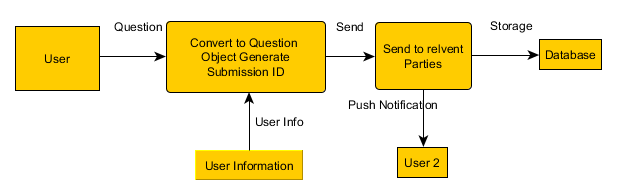
Users can generate two kinds of submissions: answer submissions, and question submissions.

**QuestionSubmission**

A question takes text and categories from the user and creates a **QuestionSubmission** object. The user enters the title for their and their questions body then presses submit. The system requests categories for the question such as course information through an alert. After the user confirms there categories a variety of information is passed to the backend of the system.

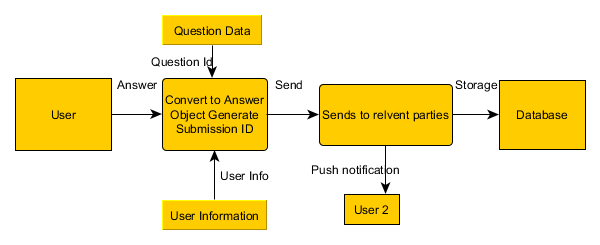
The system creates a **Question Submission** object by setting a string equal to the title and generating the remainder of a submission object which it inherits from. This involves setting a string body equal to the question text, setting an integer **AuthorID** equal to **UserID**, an integer **CategoryID** to the selected category, initializing a **Score** integer to zero and its **Status** to Active, setting a **Date JS** date objectequal to the current date. The **QuestionSubmission** also generates an empty array for **AnswerIDs**. The information is then placed in the database and sent to other users based on the **CategoryID**. The following dataflow diagram illustrates this process:



Answer Submission

An answer takes text from the user and creates an answer object. The user enters their response to a given question and presses submit. The ParentQuestionID, UserID and answer text are then sent to the back end of the system. The backend takes this information and generates an answer object.

The answer object is created by setting an integer equal to the ParentQuestionID and generating the remainder of a submission object which it inherits from. This involves setting a string body equal to the question text, setting an integer AuthorID equal to UserID, an integer CategoryID for the category, initializing a score to zero and its status to Active, setting date equal to the current date. The backend then updates the Question object in the database by placing its own Id in the array of answers found using the ParentQuestionID. Finally the backend pushes a notification to the user who asked the question by referencing the AuthorId on the question submission. The following dataflow diagram illustrates this process.



Reporting

A report takes one of two button presses from the user to generate a report and a brief description of why they are reporting them. The user selects either spam or language for the reason of the report. The backend then creates an AbuseReport object.

The backend generates an AbuseReport id for the object by setting an integer equal to the user id, setting a string reason to either spam or language, setting a string description to the description entered by the user, setting an integer SubmissionID to the reported objects submission id, setting its status to active and setting an integer ReporterId. Once the report has been generated the status of the flagged item is set to under review. The report object is then place in storage to be reviewed by an admin later. The following dataflow diagram illustrates this process.

