**Task for Artak Abrahamyan**

**Introduction**

SolarSystem(SS) needs a platform for managing different events between his departments.

Currently SS consists of 5 departments:

1. HR /4 employees/

2. Development /15 employees/

3. DevOps /4 employees/

4. Sales /6 employees/

5. Management /5 employees/

Periodically each department generates and publishes some events. For example, on behalf of HR department should be send a “Happy Birthday and Wishes ….” notification to people who have birthdays. HR notifications should be received by all and be sent automatically.

After each successful release DevOps department (usually it happens twice per day) publishes

events for Development and Management (“Congrats, Next best version of SolarSystem on

production).

Development publishes events for Management “Next Best feature is done… ready for

deployment, Your Best Team”.

Every day the sales department publishes new announcements about new deals for

Management and Development departments.

Every Manager can login and set a scheduler for some event to be sent on time automatically for all employees.

**Platform Requirements**

1. Product must have a name.

2. Product must automatically load all organization structure and employees to the system

after setup, once. If any database in use, must be some DB migrator.

3. Employees must login into the system using email. After login system shall show the

user's affiliation with the department.

4. Users must be able to change password after login.

5. Users must receive all notifications, depending on the permissions (departments)

6. Users can enable /disable notifications by departments.

7. Managers have a dashboard for managing organization structure (update account,

change department). If the account’s department is changed then other managers must

be notified.

9. Each account has a dashboard where he/she can view notifications list received per

Department, sorted by descendant order (received date)

10. Solution should be scalable and extensible.

**Technical Requirements**

Architecture: Application design and logic up to candidate.

Frontend language: up to candidate

Backend language: up to candidate

Database: up to candidate

Deployment: up to candidate. But deployment should be in place. Using Docker is preferable.

**Important aspects for review**

● Application should be compliant with the abovementioned description.

● Application should be in working state.

● Application should be deployed on client’s side easily.

● Architecture and design patterns are very important for the stockholders.

● Candidate should be ready to justify every aspect of implementation.

**Deadline:** preferable, 3-5 days after task’s assignment.