

Στο BMCS σύστημά μας, υπάρχουν τέσσερα smart contract, τα οποία είναι τα εξής:

- **Contract UserInformation:** Διαχειρίζεται την εγγραφή του χρήστη στο σύστημα, για requesters και workers. Ο τύπος του χρήστη καθορίζεται από ένα enum που έχει δύο τιμές: 0 για Worker και 1 για Requester. Οι πληροφορίες ενός χρήστη εισάγονται σε struct *User_Registration* ως εξής: uint u_id; string full_name; User_Type user_type; address user_address; uint zone; uint cancelled_tasks; uint completed_tasks; uint limit_tasks; uint reputation; . Επίσης, ορίζεται ένας πίνακας uint256 ως u_ids, ο οποίος θα περιέχει όλα τα μοναδικά ids των χρηστών του συστήματος. Οι πληροφορίες του κάθε χρήστη εισάγονται σύμφωνα με τη συνάρτηση *setUser_Information*. Η συνάρτηση *Return_UserData* επιστρέφει όλες τις πληροφορίες για έναν συγκεκριμένο χρήστη σύμφωνα με το μοναδικό user_id. Η συνάρτηση *Arrange_Limit* καθορίζει για τον κάθε χρήστη, σύμφωνα με την τιμή του reputation του, το πλήθος των task που μπορεί να αναλάβει ταυτόχρονα, αν πρόκειται για Worker, ή των task που μπορεί να αναθέσει στο σύστημα αν είναι requester. Έπειτα ορίζονται functions για να αλλάζουν τον μετρητή των tasks που ένας χρήστης έχει ολοκληρώσει (completed) ή έχει ακυρώσει (cancelled). Η συνάρτηση *Fetch_and_Set_Timezone* επιστρέφει το timestamp του block. Τέλος, η συνάρτηση *CheckUserType* ελέγχει για τον τύπο του χρήστη, δηλαδή αν είναι worker ή requester.
- **Contract Task Initialization:** Κληρονομεί το *UserInformation* και διαχειρίζεται την δημιουργία των task από τους Requesters και την εισαγωγή των διαφόρων παραμέτρων και πληροφοριών που χαρακτηρίζει το κάθε task. Οι πληροφορίες του task εισάγονται στο struct taskCreation: address user_address; string requester_name; uint256 requester_id; string task_name; uint256 unique_taskid; string task_information; int time; uint zone; TaskStatus status; int number_of_workers_limit; uint reward; . Η κατάσταση ενός Task καθορίζεται από ένα enum με όνομα TaskStatus το οποίο μπορεί να πάρει τις εξής τιμές: Available - 0, Reserved - 1, Cancelled - 2, Completed - 3. Το unique_taskid του κάθε task υπολογίζεται από την συνάρτηση *Set_Task_Unique_ID* η οποία επιστρέφει το μέγεθος της συνάρτησης task_ids, η οποία αυξάνεται κάθε φορά που προστίθεται ένα task. Οι πληροφορίες για το κάθε task ορίζονται από τη συνάρτηση *setTask_Information*. Η συνάρτηση *set_status* ορίζει το task ως Available κατά τη δημιουργία του. Επίσης, η συνάρτηση *Task_Event_Creation* ορίζει ένα event που ανακοινώνει ότι δημιουργήθηκε ένα task εμφανίζοντας την διεύθυνση του Requester που το δημιούργησε, το όνομα του task και την μοναδική id του.
- **Contract Task Selection Process:** Το contract αυτό κληρονομεί τα στοιχεία των *UserInformation* και *Task Initialization*. Σε αυτό το contract πραγματοποιείται η επιλογή των κατάλληλων Worker για το κάθε task. Ορίζεται ένα struct *Task_has_Workers* στο οποίο υπάρχουν δύο μεταβλητές, μία για το μοναδικό id ενός task και ένας πίνακας των Worker που έχουν επιλεγεί για αυτό το task. Η συνάρτηση *get_Task* επιστρέφει το task που έχει ζητηθεί(προσωρινό). Η συνάρτηση *Select_Worker* επιλέγει τους κατάλληλους workers για κάθε task σύμφωνα με ορισμένες προϋποθέσεις. Η συνάρτηση *Table_Task_Requester* εμφανίζει τις πληροφορίες των task που έχει δημιουργήσει ένας Requester και η αντίστοιχη

συνάρτηση *Table_Task_Worker* εμφανίζει τις πληροφορίες των task που έχει αναλάβει ένας Worker. Η συνάρτηση *Show_Task_Infortmation* εμφανίζει το όνομα και την περιγραφή του task που έχει ζητηθεί στις παραμέτρους. Η συνάρτηση *Task_Cancelled* καλείται όταν ένας χρήστης, Requester ή Worker, θέλει να ακυρώσει ένα task. Ανάλογα με το είδος του χρήστη, ακολουθούνται συγκεκριμένες διαδικασίες για την ακύρωση. Η συνάρτηση *getWorkerCount* επιστρέφει τον αριθμό των Worker που έχουν αναλάβει ένα συγκεκριμένο task. Η συνάρτηση *TaskStatus_Update* αλλάζει το status ενός task σε μια νέα κατάσταση.

- **Contract Reward_Penalty_System:** Πραγματοποιεί έλεγχο σχετικά με την ποιότητα των δεδομένων που έχουν υποβληθεί από τους workers και ανάλογα με αυτή στέλνει την ανάλογη επιβράβευση ή τιμωρία και υπολογίζει το ανανεωμένο reputation όλων των χρηστών.